Massachusetts

COMPUTER ASSOCIATES

division of

APPLIED DATA RESEARCH, INC.

197

MASSACHUSETTS COMPUTER ASSOCIATES
A DIVISION OF APPLIED DATA RESEARCH, INC.
LAKESIDE OFFICE PARK WAKEFIELD, MASSACHUSETTS 01880 (617) 245-9540

THIRD SEMI-ANNUAL TECHNICAL REPORT

Part II (Covering Task Area II)
(22 June 1969 - 21 December 1969)

FOR THE PROJECT

"RESEARCH IN MACHINE-INDEPENDENT SOFTWARE PROGRAMMING"

Principal Investigators:

Task Area I	Carlos Christensen	(617) 245-9540
Task Area II	Anatol W. Holt	(212) 244-5700

Project Manager;

Peter C. Waal (617) 245-9540

ARPA Order Number - ARPA 1228
Program Code Number - 8D30

Contractor:	Massachusetts Computer Associates Division of ADR
Contract No.	DAHC04-68-C-0043
Effective Date:	21 June 1968
Expiration Date:	30 September 1970

Sponsored by
Advanced Research Projects Agency
ARPA Order Number - ARPA 1228

Monitored by
Army Research Office - Durham ARO-D

CA-7004-1711
April 17, 1970

EVENTS AND CONDITIONS

An Approach to the Description

and

Analysis of Dynamic Systems

by

Anatol W. Holt

Frederic Commoner

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by U.S. Army Research Office - Durham, Box CM, Duke Station, Durham, North Carolina - 27706, under Contract DAHC04-68-C-0043.

TABLE OF CONTENTS

CHAPTER I. INTRODUCTION

INTRODUCTION

This report covers work on Task Area II of the ARPA sponsored project "Research in Machine-Independent Software Programming" covering the six-month period ending on December 21, 1969. Although technically, this report is the sequel to "The Final Report for the Information System Theory Project"[1], it is self-contained.

This report includes the work of Dr. Anatol W. Holt and Mr. F. Commoner. During the period noted above, consultations were held with Mr. Robert M. Shapiro, Dr. Carl Adam Petri and Dr. Shimon Even.

This introduction is divided into three parts. First, a summary of the objectives of the project as a whole; second a reported contract period; third, a view of things to come.

---

[1]See bibliography for this and other directly related work.

A. Summary of Objectives

There has, in the last years, been an ever widening practical and academic interest in system problems which become prominent when the system consists of many concurrently operating components -- hardware or software -- and the system environment consists of many concurrent users. The questions which arise are usually very difficult to state exactly enough so that one could speak of definitive answers, let alone actually find such answers. Here is a sample list, to give the flavor.

A1. Does the system perform the functions which are expected of it?

A2. Might the system "die" -- i.e. be brought into a state of deadly embrace?

A3. Can the system be extended or contracted by the addition or removal of system resources?

A4. Where are the system bottlenecks and what improvements in performance might be expected as a result of adding resources at bottleneck points?

A5. Suppose we wish to replace one system component by another one. Exactly what are the requirements which the new component must satisfy in order to insure "proper functioning" of the whole?

The aim of our project is to build a theoretical foundation which will make it practically possible to ask questions such as these exactly, and find definitive

answers to them.

We are entirely concerned with systems whose part-by-part performance can be thought of in discrete terms. Not pipe systems through which fluids flow, but algorithms or circuits which perform digital functions, or manufacturing units whose inputs and outputs are discrete lumps.

We are entirely unconcerned with such distinctions as hardware/software or computer/automobile factory, because the questions which we wish to ask and answer apply equally on either side of such boundaries.

There are many existing methods for the description and analysis of discrete systems. To give perspective to our work it is helpful to contrast its methods and intent with other techniques.

To begin with there are system simulation languages (like GPSS or SIMSCRIPT). In contrast to these, our style is mathematical. Our descriptive primitives are very few in number and the interesting properties of described systems are to be found by algorithms based on theorems, not by simulation. As already mentioned above, we expect the ability to formulate interesting questions in a way which admits of exact answers, not answers by statistics.

In contrast to automata theory, we are fundamentally interested in concurrent operation. This makes it impossible to build one's models on the notion system state. While I am typing at an input terminal, a processor is

compiling a program and a channel is transmitting pages from a disc to a buffer area. The notion total system state is a road block to the fruitful description of this set of circumstances. Also in contrast to automata theory is the content of our theorems. We are not concerned with questions like, what class of functions are computable by what class of devices, but rather with questions like, how far out-of-step can this part get relative to that part?

Perhaps we are closest in spirit to operations research techniques, but with an insistence on conceptual economy and rigor more common in purer branches of mathematics. Also, it is necessary that our descriptions be built up part by part in analogy to the way in which the systems being described are built up part by part. This is in contrast to many descriptions in the form of sets of equations or inequalities. Each equation or inequality usually expresses some constraint on the whole and does not correspond to a functional component. In this respect, our descriptions will resemble programming languages which allow assembly of parts.

B. Accomplishments of the Contract Period

The basis upon which the work began was the "Final Report for the Information System Theory Project". It became clear that a direct approach to the analysis of occurrence systems was too difficult and we backed off

to the study of two simpler classes of structures -- marked graphs and state transition diagrams. Both of these are special cases of occurrence systems. We have reason to hope that our developing ability to analyze these two classes will give us the tools with which to attack the analysis of systems which are Petri-net describable. Marked graphs and state transition diagrams isolate two aspects of system description from one another: the aspect which has to do with flow, and the aspect which has to do with function. The analysis of flow (marked graphs) shows where items flow and what other items they meet; the analysis of function shows the structure of the items and how they affect one another.

In the area of marked graphs effort was divided into two parts: semantics and mathematics. Here "Semantics" means developing techniques for expressing meanings about systems in marked graph form (Chapters II and III of this report). On the side of mathematics many theorems and algorithms were developed which have significant system interpretation (Chapters IV and V).

In the area of state transition analysis we developed a new technical concept of information which makes it possible to measure information quantities that flow in and out of a state machine, as well as identify the information content which flows in and out at different state transitions (Chapter VI). This work is, thus far, of

theoretical rather than practical interest.

C. Things to Come

In regard to marked graphs, we are approaching the point where it will be useful and necessary to build a program package for the construction and analysis of such graphs. Without such a package we will not learn how to build marked graph representations of practical systems. An analogy can be made to computer programming. Marked graphs (and more generally, Petri nets) are to system description as computer code is to programming. In order to represent large problems, one first needs assembly techniques, and subsequently higher-level languages. Efforts to mechanize such assembly of big descriptions are worth making since there now exist analytic tools which will reveal interesting properties of the resulting descriptions.

Several next steps are indicated in moving toward the analysis of Petri nets. One direction is the direct extension of present marked graph results. Another direction is to form the appropriate connections between our existing work on state transition diagrams and marked graphs. Both of these directions will be pursued in the next period.

CHAPTER II. SYSTEM DESCRIPTION

SYSTEM DESCRIPTION

Of what does a system consist? For example, should we take processors, inputs and outputs as the elementary entities of which they are made? Do thay have states? Do they take space (or is it only their realizations which do)? When they operate, do they take time? Etc., etc.

We shall introduce a set of elementary entities and elementary notations with which to structure, describe, and analyze systems. Formally, these entities would be presented in a set of axioms -- just as with points and lines in geometry -- here, only by an informal description.

Our starting points are the notions condition, event, and their instances. An instance of an event is called an occurrence of that event. An instance of a condition is called a holding of that condition. Out of them we hope to build the concepts with which a wide class of systems can be usefully described, categorized, and analyzed.

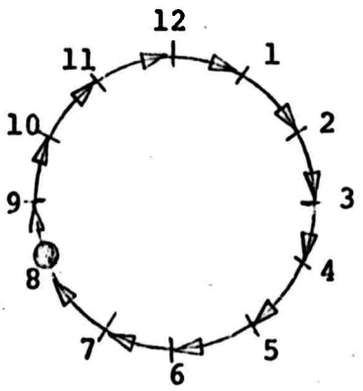
The notions condition and event have, on the face of it, several appealing properties. They have as wide a range of interpretation -- from micro levels to macro levels -- as do more standard concepts used as starting points in system description (like value, functions, storage, etc.) Also, conditions and events are by nature dynamic, temporal. They don't have to be "pepped up", as functions and values must be, in order to exhibit motion.

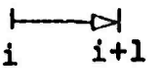
Of course, if our project is to succeed we must (a) demonstrate that all of the usual notions (including functions and values) can be reconstructed and (b) that the effort is repaid by new insight and analytic power.

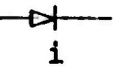
A. A First Example

We begin with a simple example of structured conditions and events.

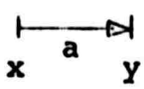
A1.



In this picture, each directed interval 

represents a condition which we can express in English: it is i o'clock. Each dividing mark 

(or vertex) represents an event (i o'clock). The picture also establishes two relations between conditions and events:

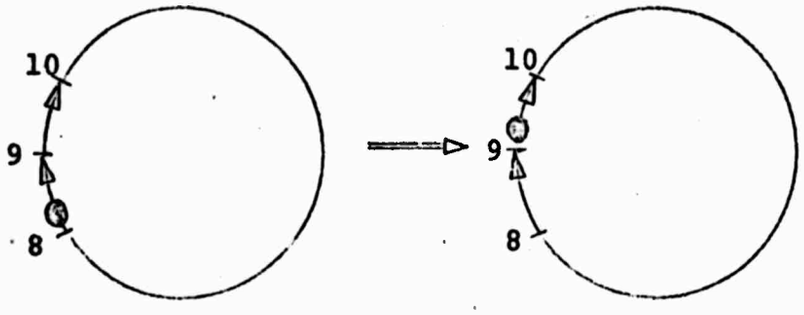


All holdings of condition a begin with occurrences of event x, and all holdings of condition a end with occurrences of event y.

In example A1 we see a small circle called a token on the 8 o'clock interval. This is a method of exhibiting

a particular holding of the condition it is 8 o'clock. By moving the token across an event symbol we can exhibit the effect of the event.

A2. The effect of the event 9 o'clock.

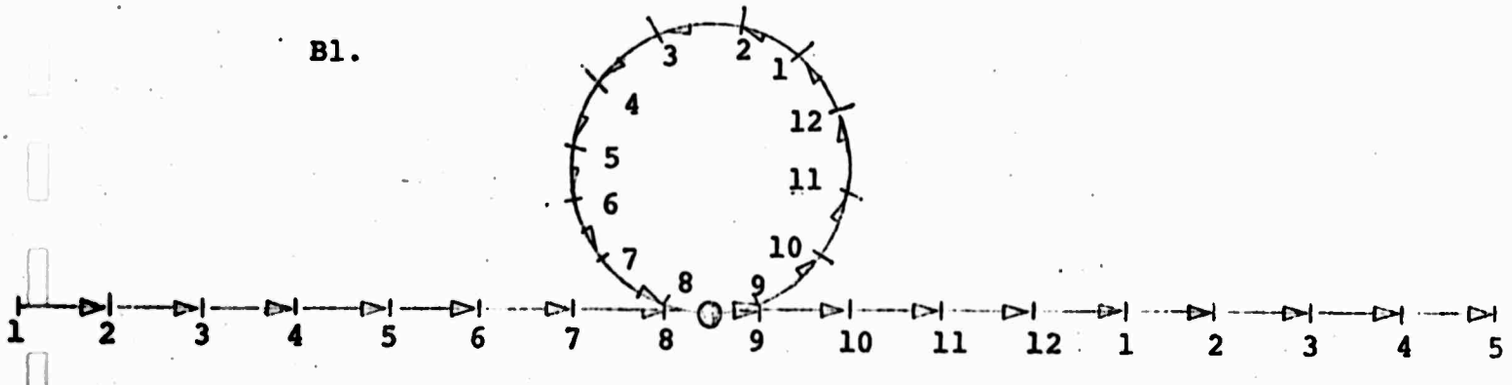


An occurrence of the event 9 o'clock ends a holding of the condition it is 8 o'clock and begins a holding of the condition it is 9 o'clock.

B. Occurrences and Holdings

Figure B1 shows the relationship between the events and conditions pictured on the clock circle, and their occurrences and holdings.

B1.

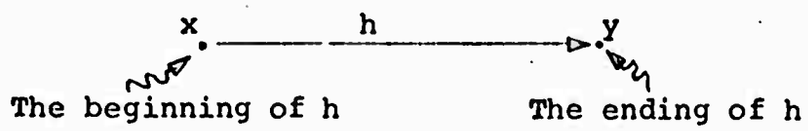


Each vertex on the line which extends indefinitely in both directions represents an occurrence and each directed

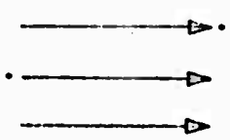
segment represents a holding. The labelling on the line shows what occurrences belong to what events and, by implication, what holdings to what conditions on the "clock". As we see, each clock event is an infinite class of occurrences and each of its conditions is an infinite class of holdings.

We can think of the occurrences and holdings in A3 as generated by the clock circle: by rolling it forwards and/or backwards. This motion is also representable by moving the token around the circle.

We will take two relations between holdings and occurrences as fundamental: occurrence x <u>begins</u> holding h and occurrence y <u>ends</u> holding h .



We will allow holdings without beginnings or without endings, or both, represented thus:



If a holding h has a beginning we denote it by $\cdot h$
If a holding h has an ending we denote it by $h^{\cdot}$

For an occurrence x , we use $\cdot x$ to denote the set of holdings h , such that $h^{\cdot} = x$; we use $x^{\cdot}$ to denote

the set of holdings h such that $x = \dot{}h$. We will also apply the dot notation to sets of holdings or sets of occurrences, to mean all elements obtained by applying the dot to the elements of the sets.

Consider an arbitrary collection of holdings and occurrences. We can represent the collection as a directed graph with the unusual understanding that there are arcs which lack a head vertex or a tail vertex or both. Now relative to a given collection S of holdings and occurrences we define:

B2. For $s_1, s_2 \in S$, s_1 is before s_2 (s_2 after s_1) if there is a directed path from s_1 to s_2. If s_1 is before s_2 or s_2 is before s_1 then we say that s_1 and s_2 are ordered. If s_1 and s_2 are not ordered then they are concurrent.

Finally we assume as an axiom:

B3. For $s_1, s_2 \in S$, if s_1 is before s_2 then s_2 is not before s_1. In other words S, as a graph contains no circuits.

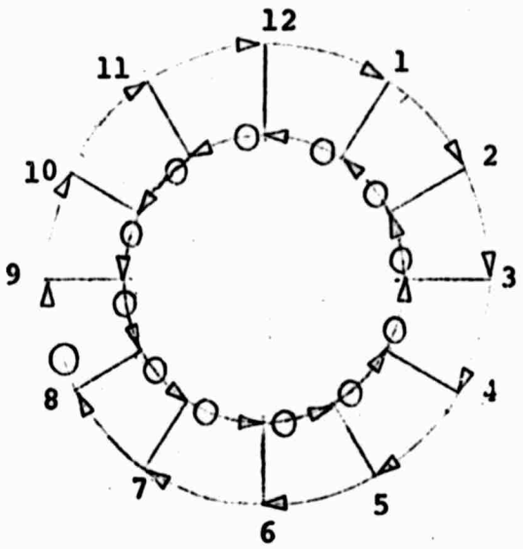
C. Concurrent Holdings

An important aspect of describing complex systems is the ability to represent states of affairs defined by the concurrent holding of many conditions. We will now show by example how this is done.

Let us picture a clock with more structure than A1.

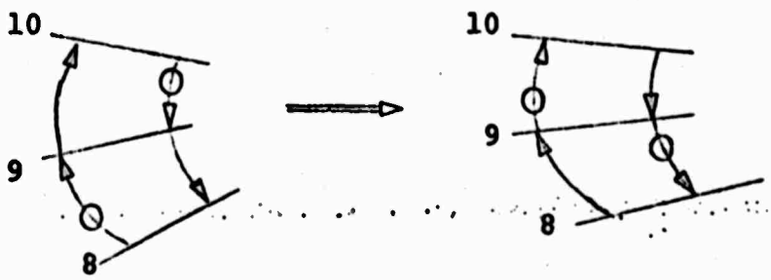
Think of the clock face as divided into 12 sectors. There is also a pointer which advances from sector to sector. The behavior of each sector is now characterized in terms of two conditions: <u>the pointer is in the sector</u>, <u>the pointer is not in the sector</u>. The behavior of the pointer is characterized by 12 conditions of the form: <u>the pointer is in sector i</u> .

C1.



In C1 we see a set of 12 <u>concurrent</u> holdings (12 tokens on arcs). Each occurrence is now the ending for two holdings and the beginning for two holdings. Correspondingly, the effect of an occurrence can be pictured by transporting two tokens across an event symbol in the diagram.

C2.

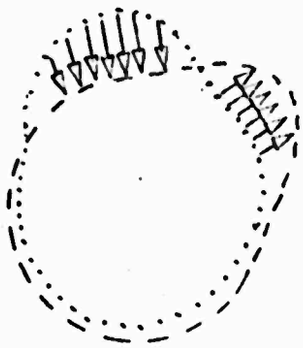


There are two other useful pictures of occurrences to consider.

First, we can think of an occurrence as expressing a <u>difference between sets of holdings</u>.

C3.

.1

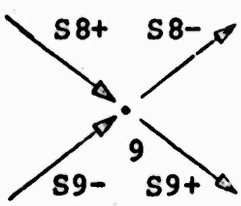


The dash set
The dot set and
The difference

.2

The occurrence which transforms the dot set into the dash set.

Contracting the line  in C3.2 to a point and labelling the arcs so as to picture the occurrence in C2, we get:



S8+ the pointer is in sector 8

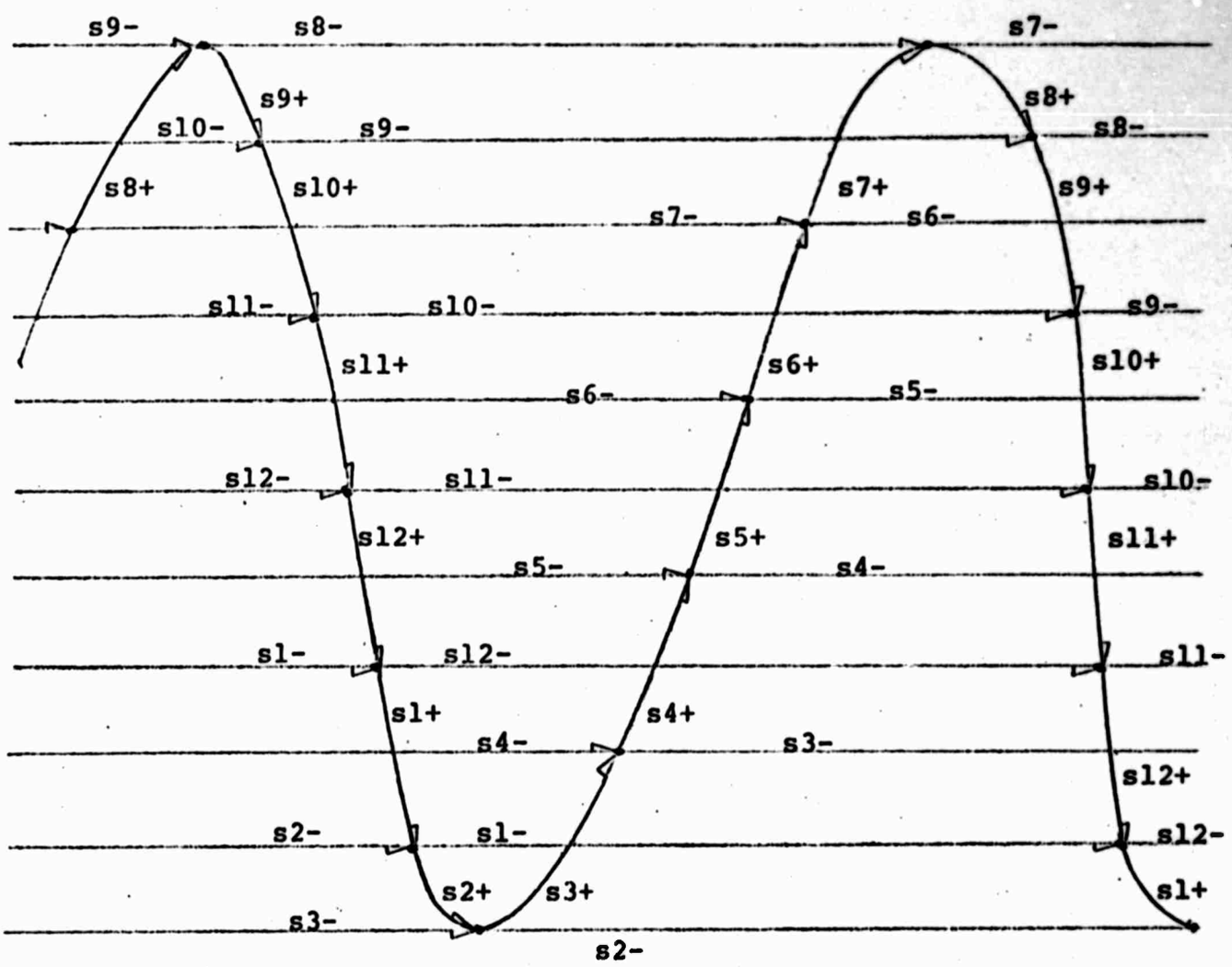
S8- the pointer is not in sector 8

S9+ the pointer is in sector 9

S9- the pointer is not in sector 9

Here is a larger picture of interconnected holdings and occurrences represented by C1.

C5.



From C5 we see at a glance that the occurrences relate to one another just as they did in B1 -- namely, there is an endless sequence of them which repeats a cycle of events.

Since we are about to consider pictures in which occurrences as well as holdings may be concurrent, it is now appropriate to express exactly what we mean by ordered holdings and occurrences with reference to pictures like C5.

C6. Two holdings or occurrences are ordered if they

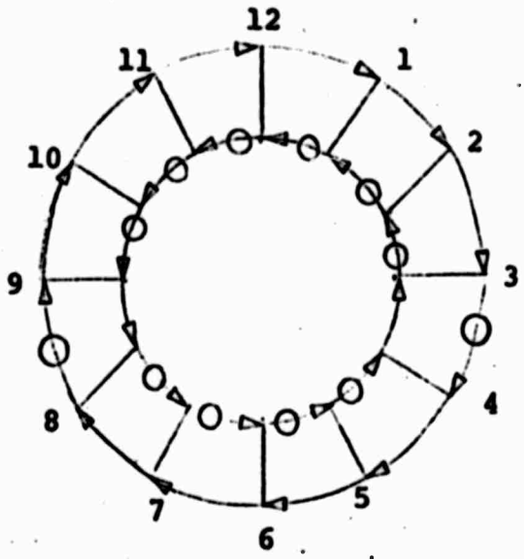
are connected by a directed path. They are then ordered in the sense of the path.

D. Concurrent Occurrences

Another inportant aspect of system description is the ability to express, formally, meanings such as: x happens while y happens -- in other words, that an occurrence of x and an occurrence of y are not ordered with respect to one another.

We get an example of unordered occurrences with a variation on C1. Imagine a clock face with two pointers, both rotating in the same direction but never passing one another. Assume further that, aside from their position relative to the clock face, the two pointers are not distinguishable one from the other. These assumptions give rise to the following variation on C1.

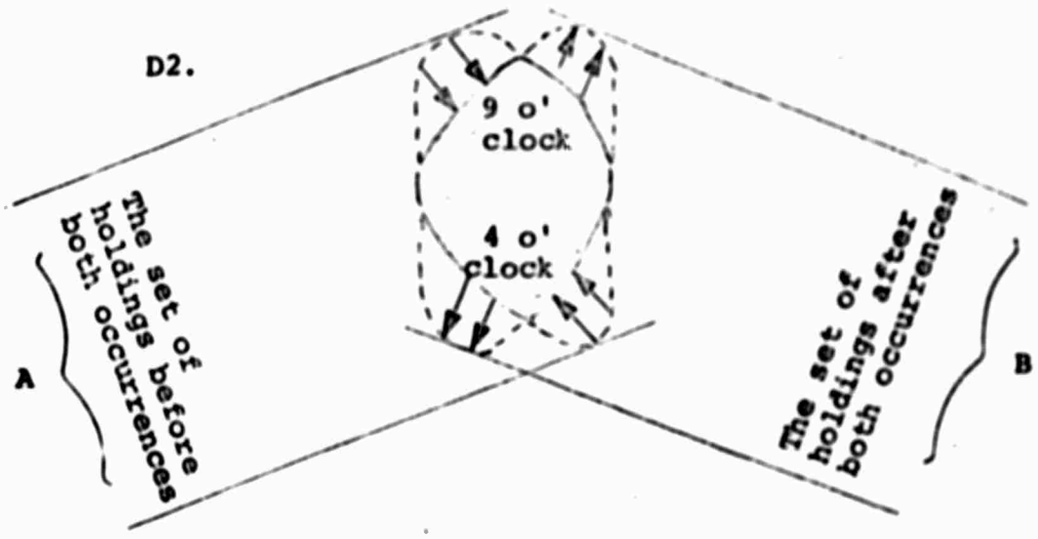
D1.



Relative to the set of holdings shown in C5 there are two

events which can occur -- 9 o'clock and 4 o'clock and they are not ordered with respect to one another -- i.e. concurrent. In the style of C3 we can picture the situation thus.

D2.



If we think of an occurrence as a set change -- knock off one "bump" and add another -- it is a change which is applicable to many different sets: namely any which has the bump which is to be knocked off and does not have the bump which is to be added. Thus we can think of the occurrence of the event 9 o'clock pictured in D2 as applied to set A, or as applied to set A already changed by the occurrence of 4 o'clock -- or, indeed, to this latter set modified yet further by an occurrence of the event 5 o'clock which is after the occurrence of 4 o'clock, and many others.

It is tempting to think of a maximal set of concurrent holdings which we call a time slice, or simply slice, as an instance of a total system state. We now see that when concurrent occurrences are possible, this picture of a

slice is seriously misleading. A total system state always stands between some last occurrence which brought it into existence and before some next occurrence which replaces it by a new one. But a slice may stand after many occurrences, any one of which could have been the last set change to bring this slice into existence. Similarly it can stand before many occurrences, each of which can be independently applied to change it.

Similarly, when there are total system states each occurrence is flanked by a definite "before/after" pair of states. But with concurrency, a given occurrence may be thought of applying to many different slices, as described above.

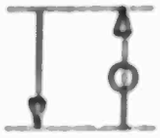
Although there is more to say about the semantics of concurrency, we now wish to examine some other aspects of C1 and D1.

E. Parts

If our claims of descriptive power are to be made good, we must show how, using conditions and events, one can construct the formal counterpart of objects which undergo change through the operation of a system. Examples C1 and D1 furnish us with some material to show how this works.

Each of these little "systems" can be decomposed into elementary parts which look like this:

E1.

.1  a sector

.2  a pointer

In both C1 and D1 there are 12 interconnected sectors and, in C1, one pointer and in D1 two of them. Each of the parts have the following notable characteristics: (a) the condition and the events are both unboundedly repeatable -- i.e. consist of infinite classes of instances; (b) every slice contains exactly one holding for each part.

We could also assemble the smallest parts as shown in E1 into larger assemblies. For example:

E2. 

A string of three sectors

Such a larger part is represented by several holdings in each slice, but still, <u>always the same number in each slice</u>.

In D1 there are two pointers. They are "identical", but have the property that at any time one can distinguish them by their sector position. In any given sector, the

two pointers appear in alternation. We would now like to show two variations on D1. In the first, the two pointers are less distinguishable than in D1; in the second, more distinguishable.

The pointers become less distinguished if we assume that they can overtake one another. That requires that two pointers be able to occupy the same sector concurrently.

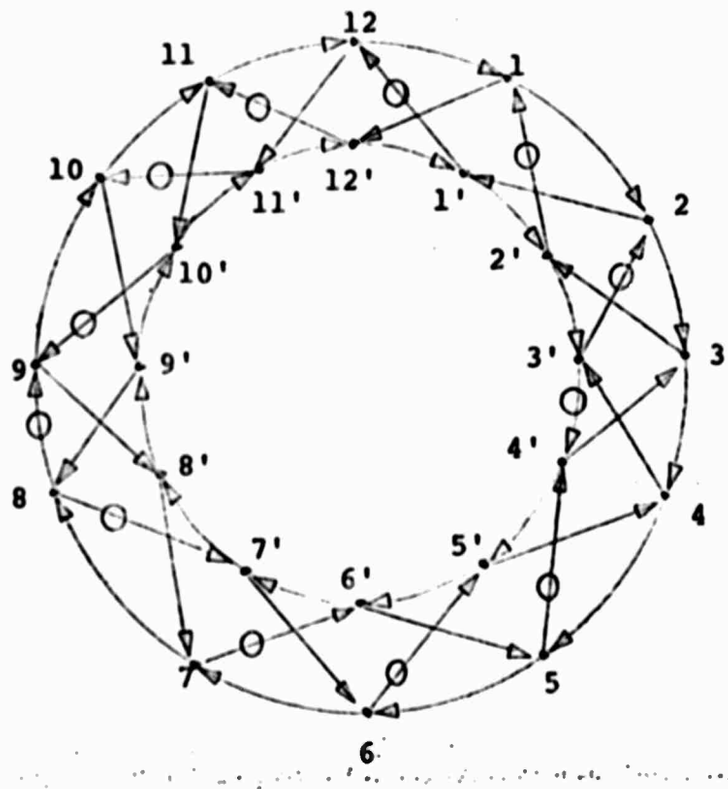
E3.

Two tokens on an arc mean that there are <u>two concurrent holdings of the same condition</u>; e.g. in E3  means a pair of concurrent holdings of the condition <u>sector i is empty</u>. Thus the sectors may be filled, twice concurrently, and may be full, twice concurrently -- i.e. occupied by two pointers.

We could look at E3 either as consisting of 12 sectors, each of capacity 2, or of 24 sectors grouped into 12 groups of 2 indistinguishable ones. Also, the slice exhibited in E3 can now be transformed by occurrences into a slice in which both tokens on the outer ring occupy the same arc. By interpretation, this is a slice in which, in respect to properties modelled in E3, the two pointers are wholly indistinguishable.

As proposed above, we shall now modify D1 so as to make the two pointers more distinguishable, this time preserving the property of D1 which prevents the pointers from passing one another.

E4.

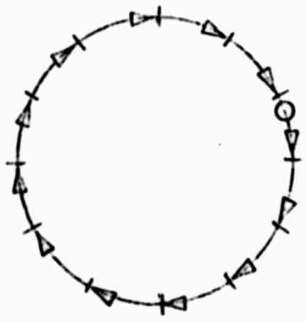


In E4 the events are represented by vertices instead of

line segments and the conditions are represented by arcs, as before. The basic parts of E4 are the following:

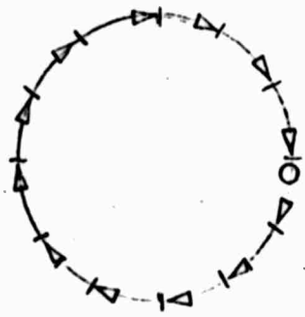
E5.

.1 Pointer 1



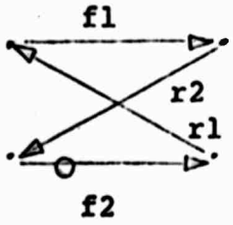
vertices named by unprimed numbers

.2 Pointer 2



vertices named by primed numbers

.3 Sector i



f1 pointer 1 is in sector i
r2 sector empty, ready for pointer 2
f2 pointer 2 is in sector i
r1 sector empty, ready for pointer 1

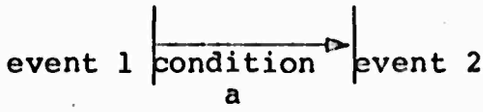
F. Events in Conflict

Thus far, we have said nothing about choice among alternative events, but the ability to express this is also fundamental. Choices among alternative events may have many meanings: whether the next input is a zero or a one, whether a given server will next serve a particular one among a set of waiting users; whether a delay will

occur or not; etc. etc. Once again we will indicate by a simple example, how such choices are expressed.

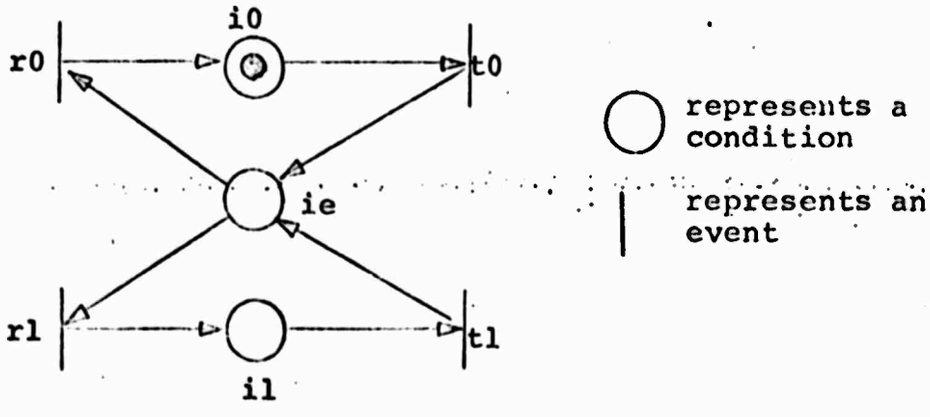
Assume that we have a sequence of "cells" each with three states, empty, zero and one. If the cell is in state zero or in state one we can say that it contains a value. If a cell has a value it will pass it to its next neighbor.

To represent this we must replace the arrow symbol which was used above for conditions by another symbol. It was true for all conditions we previously depicted that all of their holdings began with occurrences of the same event, and ended with occurrences of the same event, depicted by the drawing:



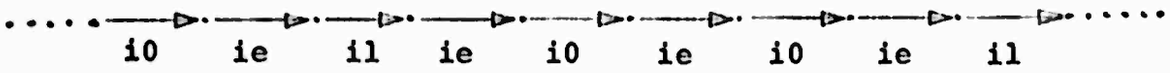
In our new example we will expect some holdings of cell i is empty to be replaced by cell i contains a zero (i0 for short) and others by cell i contains a one (i1). Similarly, cell i is empty (ie) will sometimes replace cell i contains a zero and sometimes cell i contains a one. We can picture these relations thus:

F1.



The black dot in F1 is a token which represents a holding of i0 . We could now use F1 to generate pictures of holdings and occurrences in the same way as in B1, keeping in mind that any given holding of ie can only end with exactly one occurrence, hence an occurrence of r0 or of r1 exclusively, and begins with exactly one occurrence, hence an occurrence of t0 or t1 exclusively.

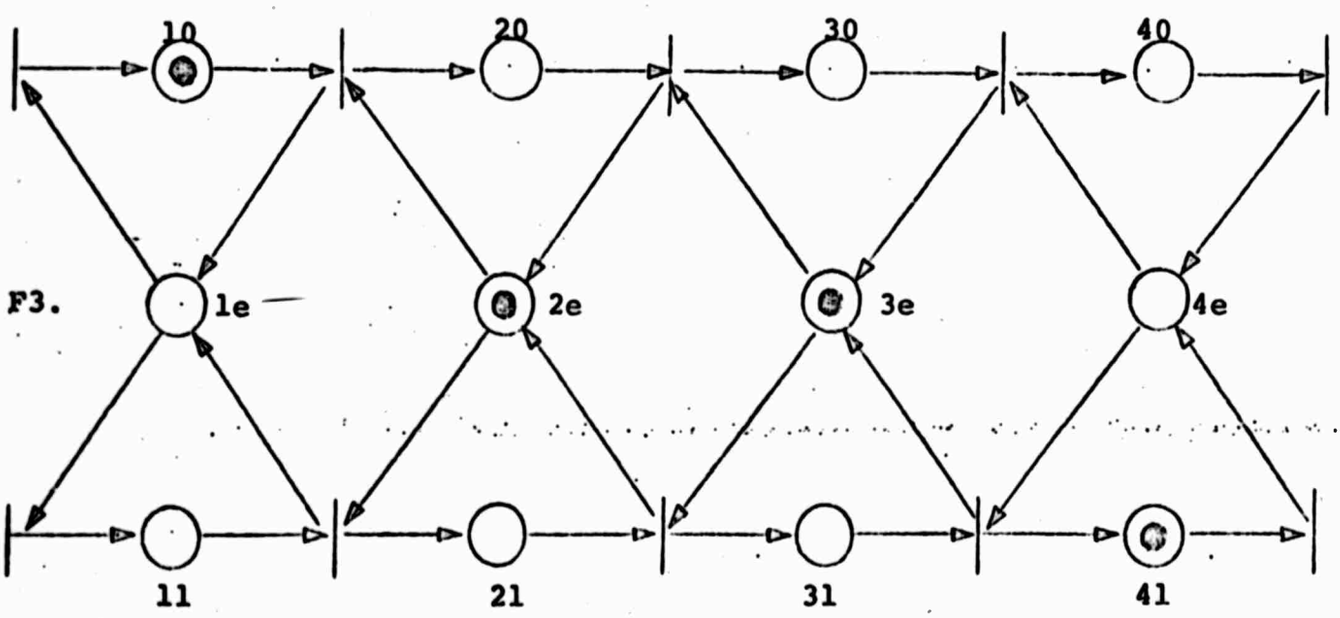
F2.



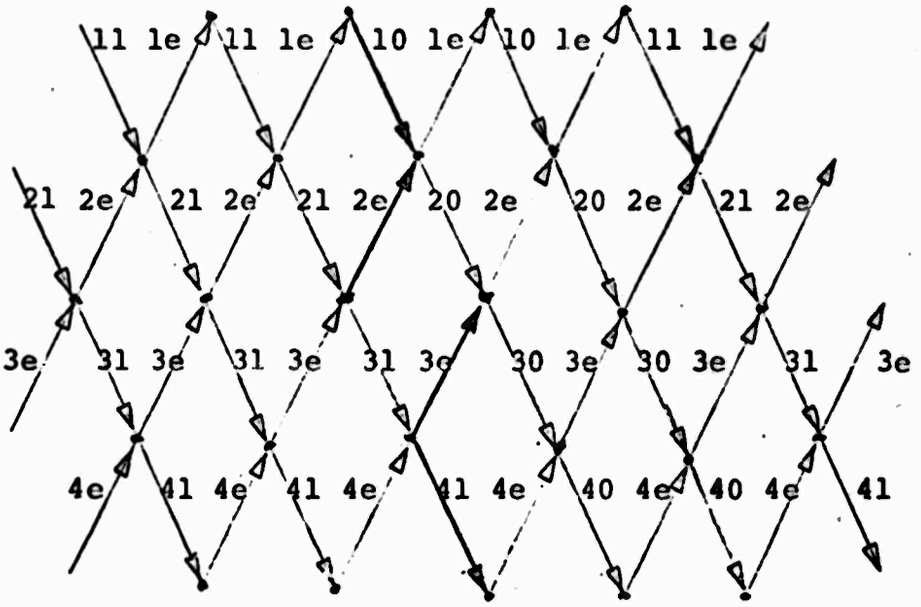
About the events in F1 we can say: r0 and r1 <u>are in forwards conflict</u> because, in slices in which ie holds, a holding of ie may end with an occurrence of the one or of the other, exclusively; <u>t0 and t1 are in backwards conflict</u> because, in slices in which ie holds, holding of ie may begin with an occurrence of t0 or t1 exclusively.

Now we can show a construction of a sequence of such cells.

F3.



F4 shows some holdings and occurrences related to F3 where the occurrences ending successive holdings of 1e were chosen arbitrarily. The dark arcs represent the slice which is shown in F3.



G. Axioms of Equivalence

Systems, according to our view, are collections of holdings and occurrences, related to one another by begin and end and grouped into condition classes and event classes. In the discussion so far we have shown two kinds of pictures, both in the form of graphs: pictures of holdings and occurrences, and pictures of systematically related conditions and events. These latter pictures included so-called tokens to show concurrent sets of condition holdings.

We are now concerned with establishing principles for the grouping of holdings into condition classes and occurrences

into event classes. Of interest to us will be groupings which are partitions. Such partitions can be thought of as equivalence relations.

G1.	.1	$h_1 \equiv_c h_2$	h_1 is condition equivalent to h_2, meaning for all conditions c, $h_1 \in c$ iff $h_2 \in c$
	.2	$x \equiv_e y$	x is event equivalent to y, meaning for all events e, $x \in e$ iff $y \in e$

Now a crucial question arises for the description of systems. What are the rules which govern translating one's knowledge of condition holding into one's knowledge of event occurrence, and vice versa? More formally expressed, how shall the notions same condition ($h_1 \equiv_c h_2$) relate to the notion same event ($x \equiv_e y$) ?

There are various sensible criteria one could introduce with various consequences for the class of system descriptions obtained. We will mention several criteria of interest to us. A thorough study of the subject introduced by such criteria is, however, beyond the scope of this presentation.

The first criterion, which we will assume as given for all system descriptions discussed in this report is:

G2. Any two occurrences of the same event end holdings of the same set of conditions (if any) and begin holdings of the same set of conditions.

We can express G2 more formally as follows. Call two sets of holdings condition equivalent if there is a 1-1 correspondence between the two sets with all matched pairs being condition equivalent. Then,

G2'. $x \equiv_e y \Rightarrow {}^{\bullet}x \equiv_c {}^{\bullet}y$

and $x \equiv_e y \Rightarrow x^{\bullet} \equiv_c y^{\bullet}$

A further set of criteria of which we will make occasional use, singly or in combination, are the following:

G3. .1 $h_1 \equiv_c h_2 \Rightarrow {}^{\bullet}h_1 \equiv_e {}^{\bullet}h_2$

.2 $h_1 \equiv_c h_2 \Leftarrow {}^{\bullet}h_1 \equiv_e {}^{\bullet}h_2$

.3 $h_1 \equiv_c h_2 \Rightarrow h_1^{\bullet} \equiv_e h_2^{\bullet}$

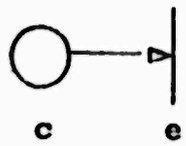
.4 $h_1 \equiv_c h_2 \Leftarrow h_1^{\bullet} \equiv_e h_2^{\bullet}$

Now suppose we use the two symbols for conditions and events introduced in F1:

conditions: ○

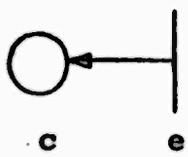
events: |

and interpret the relation



as: any occurrence of e ends one holding of c ,

and the relation



as: any occurrence of e begins one holding of c .

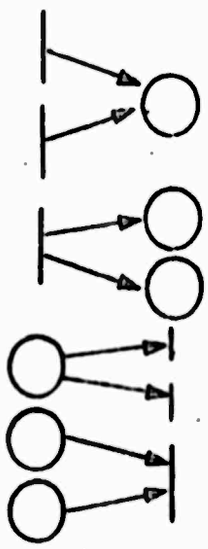
Then we can translate the criteria expressed in G3 into pictures.

G3.1 excludes

G3.2 excludes

G3.3 excludes

G3.4 excludes



G4. .1 Example A1 satisfies all four criteria, and is the only example shown which does this.

.2 All subsequent examples up to (but not including F1) satisfy G3.1 and G3.3 but not G3.2 and G3.4.

.3 F1 satisfies G3.2 and G3.4, but not G3.1 and G3.3.

.4 F3 satisfies none of the four criteria.

It would be interesting to study various existing formalized methods of describing events and conditions in regard to their "axioms of equivalence". Here we would only like to point out that everything described by a state transition diagram conforms to G2, and G3.2 and G3.4 if one interprets the states as conditions and the transitions as events. Descriptions which satisfy G2, G3.1 and G3.3 are what we call marked graphs. Descriptions which satisfy G2 only, we have called Petri nets. Thus we see that marked graphs and state transition diagrams are examples of Petri nets subject to particular (and, in a certain sense, dual) restrictions.

Crudely, one could express the difference between these two specializations thus: marked graphs allow concurrency, but no conflict; state transition diagrams allow conflict but no concurrency. Petri nets allow both.

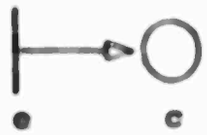
H. Petri Nets

In the preceding section, a notation was introduced for representing the conditions and events associated with a set of holdings and occurrences which satisfies axiom G2. Such a representation is called a Petri Net.

H1. In a Petri net, conditions are represented by circles: ○ called places and events are represented by bars: | called transitions. Arrows between these symbols have the

following meanings:

means: Every occurrence of event e ends one holding of condition c .



means: Every occurrence of event e begins one holding of c .

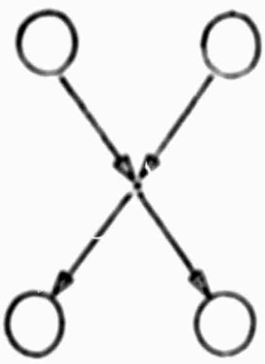
We will sometimes use arrows to represent places or transitions which, if indicated explicitly, would only have one incoming arrow and one outgoing arrow. Thus:

means:

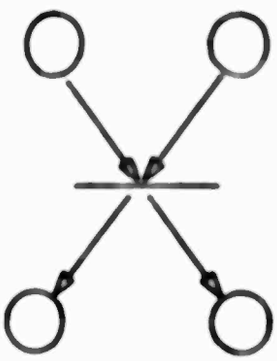
means:

Finally, we will use a dot · interchangeably with the transition symbol: | .

Thus



means



H2. If, in a Petri net, we wish to represent a holding of a condition c , we place a <u>token</u> on the corresponding place. The function which specifies the number of tokens on each place in a Petri net is called the <u>marking</u> of the net.

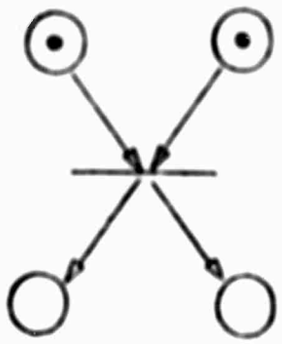
Example:

No holdings of c : 

One holding of c : 

Two holdings of c : 

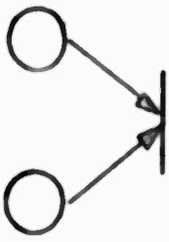
If every condition which is required for a certain event holds, an occurrence of that event takes place. This occurrence ends a certain set of holdings and begins a certain set of holdings. In a Petri net we represent an occurrence of event e by <u>firing</u> the transition t which represents e :

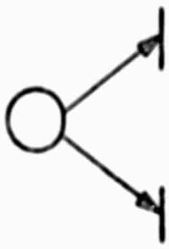


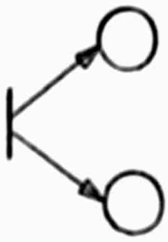
This is accomplished by removing one token from the places which have arrows pointing to t and then adding one token to the places to which t points:

H3. We are particularly interested in Petri nets which satisfy certain constraints mentioned in section G:

G3.1 excludes 

G3.2 excludes 

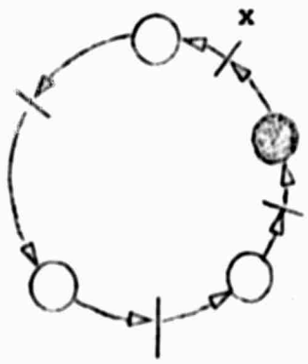
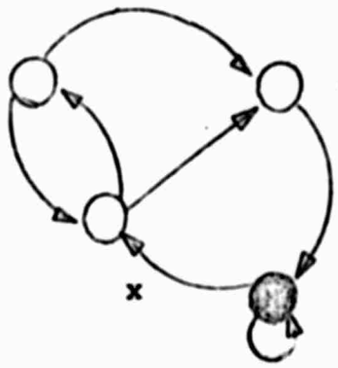
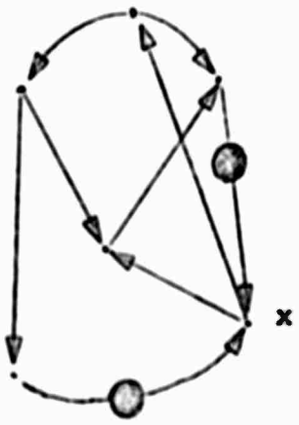
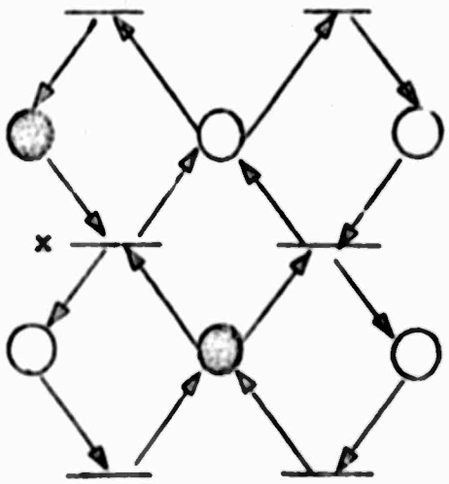
G3.3 excludes 

G3.4 excludes 

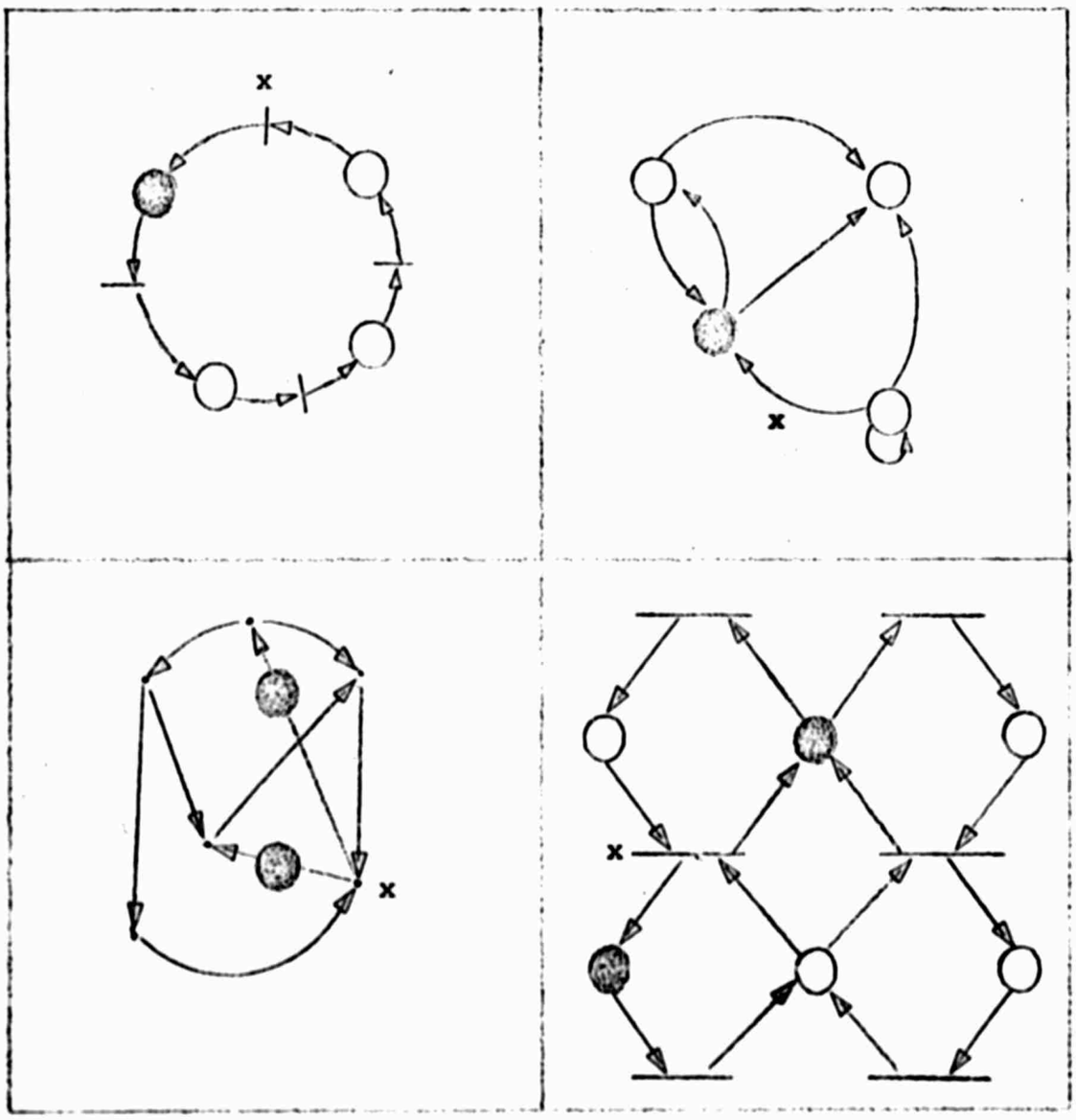
A Petri net which satisfies G3.1 and G3.3 is called a marked graph. A Petri net which satisfies G3.2 and G3.4 is called a state machine graph.

Using the conventions in H1, every place in a marked graph may be represented as an arrow; every transition in a state machine graph may be represented as an arrow.

These special types of Petri nets are illustrated in the table below:

Is it a state machine graph? \ Is it a marked graph?	Yes	No
Yes		
No		

The firing rule is applicable to all four of these Petri nets. Here are the markings after firing the transitions marked x :



H4. Marking Classes

If M is a marking of a Petri net and σ is a non-empty sequence of firings which is possible starting at M, we say M[σ⟩ exists. If the result of this sequence is a marking M', we say M[σ⟩ = M' or simply write M[σ⟩M'.

If there exists a σ such that $M[\sigma\rangle M'$ we say <u>M leads to M'</u>, written $M[\rightarrow M'$. $\vec{M}$ is the set of markings M' such that $M[\rightarrow M'$; $\overleftarrow{M}$ is the set of markings M' such that $M'[\rightarrow M$. $\overleftrightarrow{M}$ is $\vec{M} \cup \overleftarrow{M}$ and is called the <u>marking class</u> of M.

H4.1. A marking M of a net is said to be <u>live</u> if $M' \in \{M\} \cup \vec{M}$ implies that for any transition t, there exists an $M'' \in \vec{M'}$ such that t is firable in M''.

H4.2. A marking M is said to be <u>safe</u> if $M' \in \{M\} \cup \vec{M}$ implies that M' places at most one token on any place.

All the examples in section H3 are both live and safe.

If a marked graph or state machine graph is <u>strongly connected</u>*, it has a live safe marking. A strongly connected state machine has only one live and safe marking class; any marking which places only one token on the net is a member of this class.

A marked graph may have several marking classes. A procedure is given in Chapter V, Section F, page 111 for finding a live safe marking for a strongly connected graph.

---

*<u>strongly connected</u> means: There exists a directed path from any vertex to any vertex.

CHAPTER III. MODELLING PRODUCTION FACILITIES

MODELLING PRODUCTION FACILITIES

In this chapter, we will show how to use Petri nets for building models of production facilities. We will place particular emphasis on whatever can be modelled by marked graphs because we can analyze marked graphs for their interesting properties. We cannot do this yet for wider classes of nets. What is described in this chapter is only one way, among many, to build Petri nets with an interpretation.

A production schema is a description of a production facility which repeatedly converts certain inputs to certain outputs. Although it is redundant to say "repeated use of a facility" (since facilities are established for repeated use), the emphasis is important. Petri nets in general and marked graphs in particular are primarily concerned with interrelated cycles of activity, rather than finite sequences with beginnings and endings. (While in many schemes of analysis, cycles are treated by cutting them open, our techniques often make it desirable to close open stretches so as to form cycles.)

An example of a facility might be an automobile factory. Here the primary inputs are concrete objects and materials such as rolls of plastic, sheets of metal, cans of paint, boxes of rivets, etc.; the primary outputs are also concrete objects such as sportscars, sedans, panel trucks, etc.

A second example of a production facility is an assemblage of computing equipment coordinated by some combination of plug boards and/or programs to accomplish inventory control. Here the inputs are more abstract, namely symbolic expressions representing arrivals and departures of inventoried items,while the outputs are again symbolic expressions representing reports on the status of the stock.

Although both of the above examples are relatively "large" systems -- many parts and many sub-processes -- that is not a necessary feature of what could be usefully treated; the task could be to describe and analyze a facility which computes $c(a+b)$ from the inputs $a,b,$ and c.

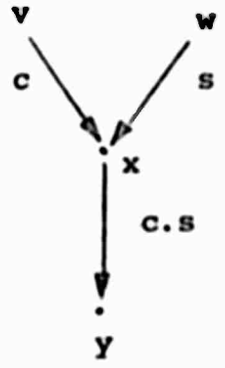
Most of the operations in the above examples can be thought of as operations of <u>assembly</u> -- i.e., the putting together of parts to make a single whole. Thus in computing $c(a+b)$, a and b are "assembled" to produce $a + b$ which, together with c, is assembled to produce the result. We are however, also interested in productions which are, in part or in whole, <u>disassembly</u>: for example, unpacking a crate which contains many packages (of packages). In computing, a disassembly could be the unpacking in memory of a complex record brought in from secondary storage. Or, it might mean the multiple copying of a single value for concurrent use by several processes.

A. The Components of Description

To construct our models, we let the events be production events, and the conditions be the states of readiness of various materials for participation in production events.

Examples:

A1.



c	a cup is now ready for x to occur
s	a saucer is now ready for x to occur
x	assembly of a cup and saucer
c.s	a cup and saucer are now ready for y to occur.

A2. Comments:

.1 Within a given cycle of manufacture there might be several distinct "times" and/or distinct "places" at which a cup and saucer are assembled. All of these would appear as separate events in the model.

.2 There are many possible interpretations for the event w (which produces the ready saucer). It might be an event which transports the saucer to a designated physical place where the assembly can take place; it might be an event which results in saucers being dry so that they can be assembled.

.3 Suppose there were two tokens on the arc representing the saucer condition. It would mean two saucers ready to participate in x . In a given production

schema this possibility may have been part of the intention, but in another, it may represent an unintentional error. Here are two reasons why it might be an error.

For two saucers to be ready would probably have to mean a saucer storage with a capacity for two-or-more saucers connected to the active agent which assembles cups and saucers. The concurrent appearance of two ready saucers would certainly be an error if the actual saucer capacity at that point in the system is one.

A second difficulty might arise from the fact that two tokens on that arc represent two ready saucers not distinguishable from one another with respect to the event x . Now if it is essential to proper production that the saucers and the cups flow through the system in strict sequence and, at the assembly point, become paired, n^{th} saucer to n^{th} cup, then two tokens on the arc must be an error. If two saucers are now ready there is no guarantee which one will be assembled with the next cup. This latter kind of reason is often of importance in models of algorithms where the cups and saucers are values being assembled by arithmetic operations. In that context it is common that the several values to be combined by the algorithm come

in related "waves" and it is an error when there is no guarantee that the next value of s meets the next value of c . As was already mentioned in the last chapter, graphs in which no condition symbol ever holds more than one token, are called safe.

A vertex with several input arcs and only one output arc represents a pure assembly operation. A vertex with several outputs but only one input represents a pure disassembly operation. There will also be vertices which have multiple inputs and multiple outputs.

Consider, for example, the following parenthesis structure:

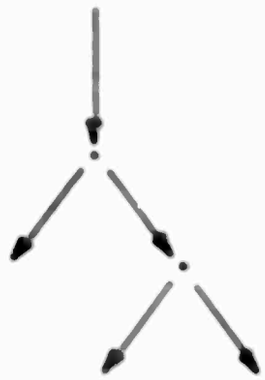
(( )(() ()))

It might represent an arithmetic expression to be evaluated by a partial ordering of assembly steps:

A3.

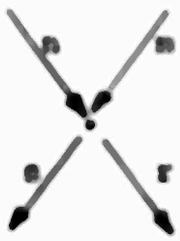
or a nested set of boxes to be unpacked (arithmetic expression to be parsed?) by a partial ordering of disassembly operations:

A4.



Here is an example of an event with several inputs and several outputs.

A5.



two natural numbers with $n < m$

compute natural numbers r and s so that $m = s \cdot n + r$, $r < n$

B. Various Flows

We will show how to build up a production schema in roughly three steps:

.1 Primary Flow

.2 Back Flow

.3 Internal Circuits

Primary Flow pictures the raw materials transformed through however many intermediate stages are necessary into the actual output packets. Back Flow pictures the flow of orders and permissions which control primary flow. Internal Circuits depict the circulation of internally moveable (re-allocatable) facilities.

C. Primary Flow

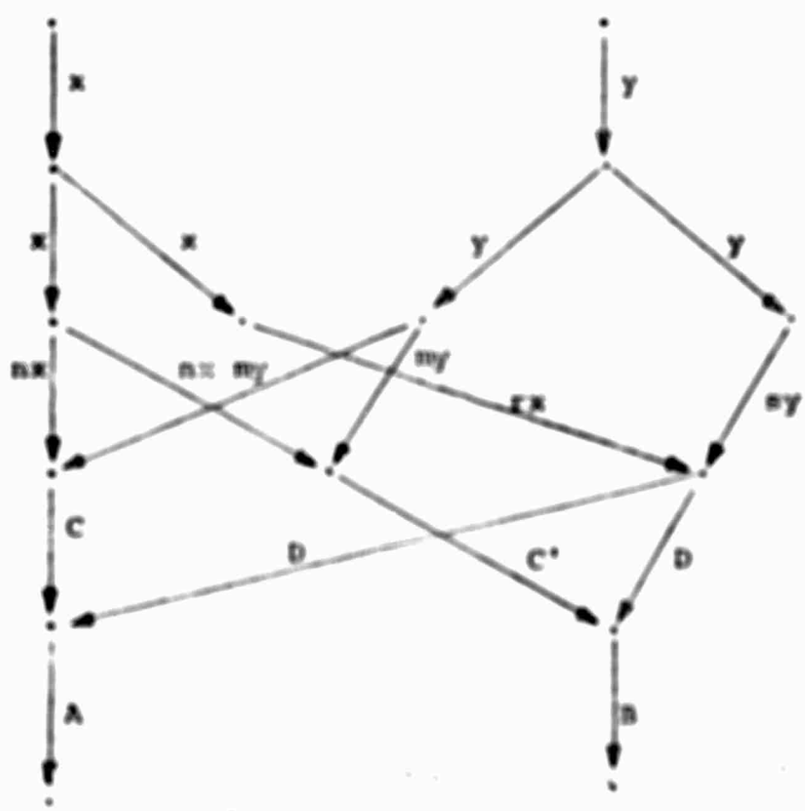
Using marked graphs only, primary flow structure will be a partial ordering. Here is a typical example.

C1.

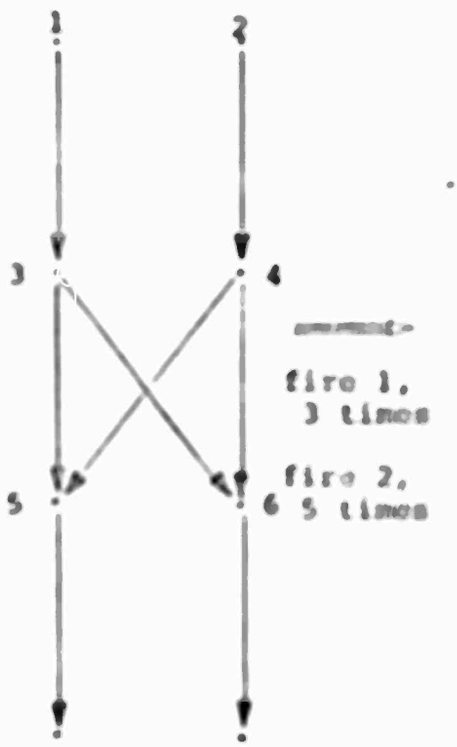
.1 A facility for computing two outputs, A and B, from two inputs, x and y, defined as follows:

$$A = \frac{\overbrace{nx + my}^{C}}{\underbrace{rx + sy}_{D}} \qquad B = \frac{\overbrace{nx - my}^{C'}}{\underbrace{rx + sy}_{D}}$$

.2 Primary Flow:



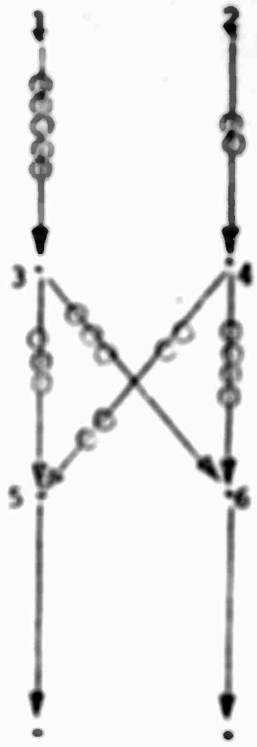
Let us see what happens if we take Cl.2 as a finished production schema for Cl.1. This means studying the properties of Cl.2 as a marked graph. By applying the firing rule to Cl.2 as a marked graph, one can advance to "later" markings by firing the 0-level vertices arbitrarily many times concurrently (or, one could move backwards to "earlier" markings by firing the level 5 vertices backwards, arbitrarily many times concurrently). Thus, using the firing rule unrestrainedly, one could "flood" Cl.2. Once there are lots of tokens on the graph, one can fire any number of vertices of the graph any number of times concurrently. Here is a partial ordering graph smaller than Cl.2 with which we can illustrate these possibilities of markings and firings.



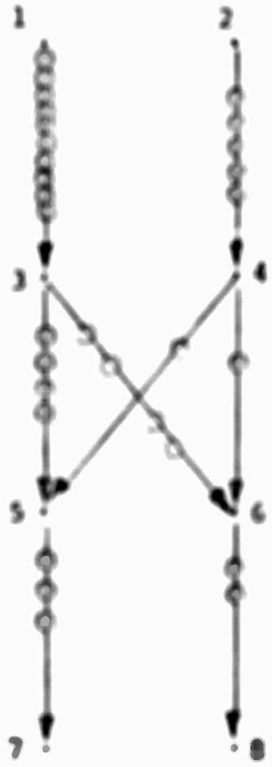
fire 1,
3 times

fire 2,
5 times

fire 1,
5 times
fire 3,
3 times
fire 4,
4 times
fire 2,
2 times



fire 1,
10 times
fire 2,
5 times
fire 3,
4 times
fire 4,
1 time
fire 5,
3 times
fire 6,
2 times



fire 1, 200 times
fire 2, 25 times
fire 3, 11 times
fire 4, 6 times
fire 5, 2 times
fire 6, 2 times
fire 7, 3 times
fire 8, 2 times

(Note: This marked graph is not safe.)

If C1.2 is interpreted by the marked graph firing rule it will behave in the way we just illustrated, and that will make it a bad model for C1.1 for several semantic reasons.

First, if one assumes that every holding requires the use of some resource (e.g. a storage unit) and every occurrence involves the use of some resource (e.g. a processor unit) then C1.2 models a facility with unbounded resources, contrary to natural assumptions about facilities. Secondly, if the construction of outputs from inputs is interpreted as arithmetic evaluation, then one should assume that successive values of x's and y's are different. In the operation of C1.2 no order is kept in the proper associations of successive x , y pairs, nor can one know which outputs relate to which inputs. (Both of these difficulties were described in general terms in A..3.)

Since pictures very similar if not identical to C1.2 are often used to represent information flow in a computation such as C1.1 without any of the strange semantic consequences which follow from interpreting C1.2 as a marked graph, it is interesting to dwell briefly on some of the differences between more conventional views of such diagrams and the marked graph view.

Still using the concepts of vertex firings and token transport, we could "operate" the graph as follows:

C3.

.1 Begin by putting one token on each of the two arcs x and y between levels 0 and 1 .

.2 Fire other vertices in any order that is possible (for example level by level).

.3 End when the tokens have arrived on the output arcs, A and B .

With the interpretation C3. beginning and ending are special procedures. Having "operated" the graph once, one can of course, operate it again by reinitializing. But the sequence constraint which prevents reinitializing until the last operation is complete is only implied by the "operating rule" C3, in contrast to the sequence constraints expressed within the diagram. After the next following discussion of back flow, it will become clear how the intent of C3 could be expressed by the marked graph firing rule.

D. Back Flow

How can one correct the defects of C1.2 under marked graph interpretation? As it turns out both problems -- modelling finite facility resources and insuring proper sequencing of items through the facility -- can be solved by the same means.

Limits can be placed on the need for concurrently available facilities by taking another class of inputs and outputs into account, namely those interpretable as orders

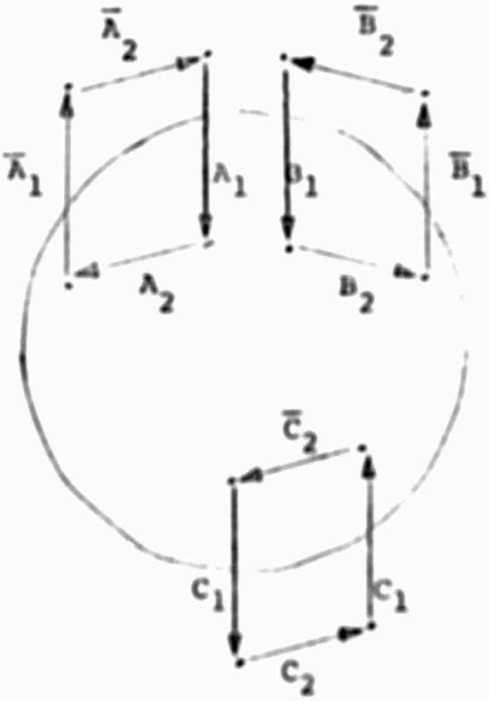
and permissions. There are orders (permissions) for output which travel from the environment to the facility, and are therefore inputs to the facility. There are also orders (permissions) for input which travel from the facility to the environment and are therefore outputs from the facility. Finally, there are orders (permissions) which are passed internally from one production event x to another one y if y is responsible for some part of the input to x .

We shall next show a series of pictures which serve to explain how the flow of orders and permissions -- back flow, for short -- help to model a production facility. These pictures will be built on the figure:

which represents some arbitrary primary flow (in the case of marked graphs, a partial ordering) with two primary inputs and one primary output.

D1.

.1



A_1 - An A-item is available to the facility

A_2 - The facility has received an A-item

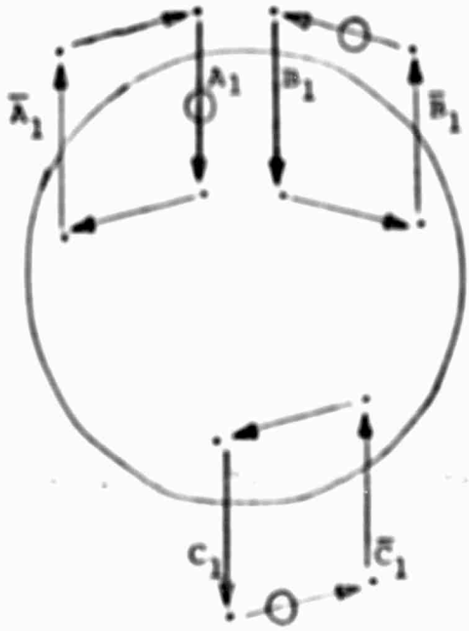
$\overline{A}_1$ - An order for an A-item is available to the environment

$\overline{A}_2$ - The environment has received an order for an A-item

and similarly for B and C .

Now let us consider the case where the environment will not submit a new order for output unless the last order has been delivered and the facility likewise will not submit new orders unless the last order for input has been delivered. We can represent this by putting one token on each of the three circuits in D1.1.

.2



We must now digress briefly to point out some fundamental facts about marked graphs

D2. In any marked graph, the number of tokens on any circuit can never be changed by vertex firings.

This is because any circuit must enter any vertex in the graph the same number of times that it exits from it. On the other hand a vertex firing takes one token from every entering arc and puts one token on every outgoing arc. Therefore in D1.2 there will never be more than one token on any of the arcs A_1, B_1 or C_1.

A circuit in a marked graph which has exactly one token on it is called a basic circuit. The three circuits in D1.2 are basic circuits. If we regard the events and conditions on a circuit as the elements of the circuit we now observe:

D3.

Any two distinct elements on a basic circuit must "alternate". For example, after a holding of B_1, and before the next holding of B_1, there must be a holding of B_2, assuming that B_1 and B_2 are on a basic circuit; and similarly for occurrences of events.

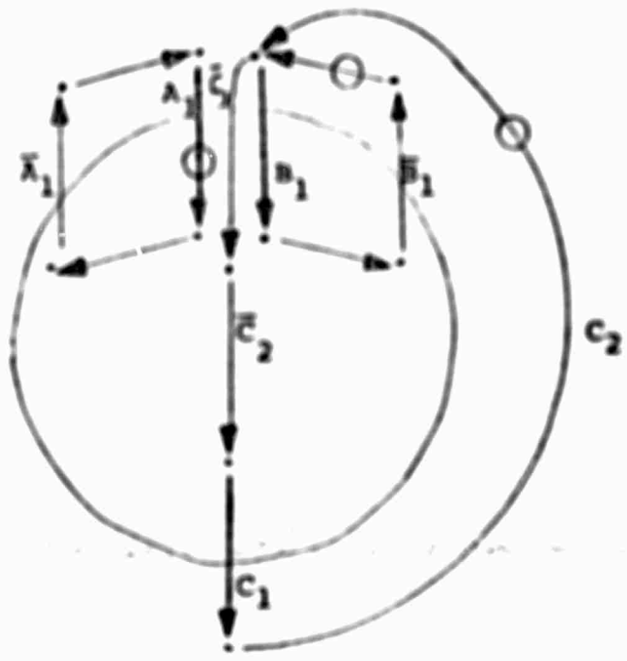
In D1.2 the environment never handles more than one order for an A-item at a time, and the facility never handles

more than one A-item at a time. Notice also that the arc labelled A , which is part of primary flow for the facility could be viewed as back-flow for the environment: the delivery of the next A-item can be interpreted as a permission given to the facility to deliver the next order for an A-item.

Consider next a modification of D1.2. Let the event which yields a B input and the event which yields a new order for output ($\bar{C}_1$) be coincident. In programming, this is the case if each call on a subroutine (the facility) is accompanied by a parameter specification which is either part-or-all of the input on which the routine must operate. It is also the case with shoe repair shops. The delivery of the order for a repair coincides with delivering the shoes to be repaired.

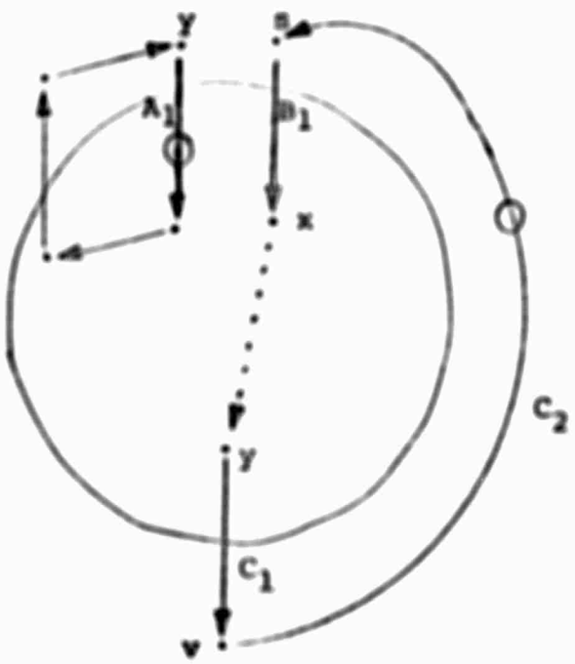
D4.

.1



Now note: if we ignore the actual handling or orders by the facility, then the facility modelled by D4.1 is the same as:

.2



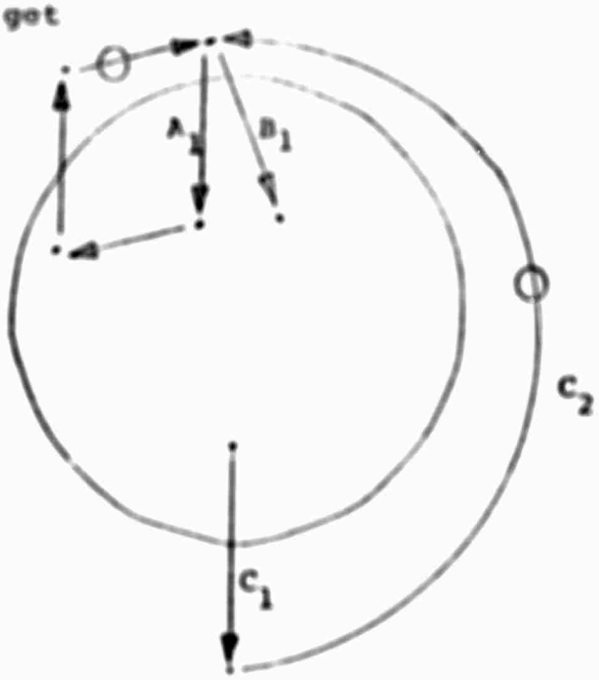
The dotted arrow in D4.2 serves as a reminder that the event x is guaranteed to be before the event y because primary flow is modelled as a partial order with y and s as its only earliest elements and v as its only latest element.

Now we can see that the effect of the arc C_2 with its token on the model is to make every primary flow path from x to y a part of a basic circuit. Therefore, by D3, any two elements on any such path -- production events or conditions -- will have to alternate. By D2, no arc on any such path will ever hold more than one token. These and only these effects on the model of the facility are also

achieved in D4.1.

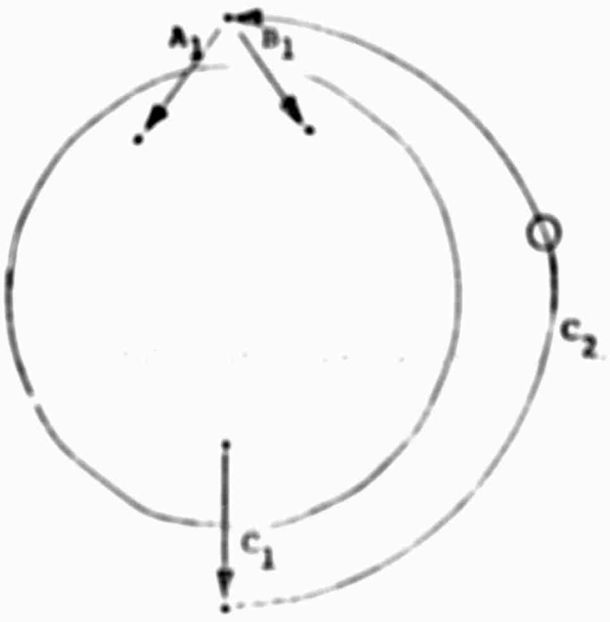
If we now further modify D4.2 in accordance with the new assumption that the events which deliver the next order for output, the next B input and the next A input all coincide -- in other words, that the order for next output is accompanied by all input items needed to make that output, then we get

.3



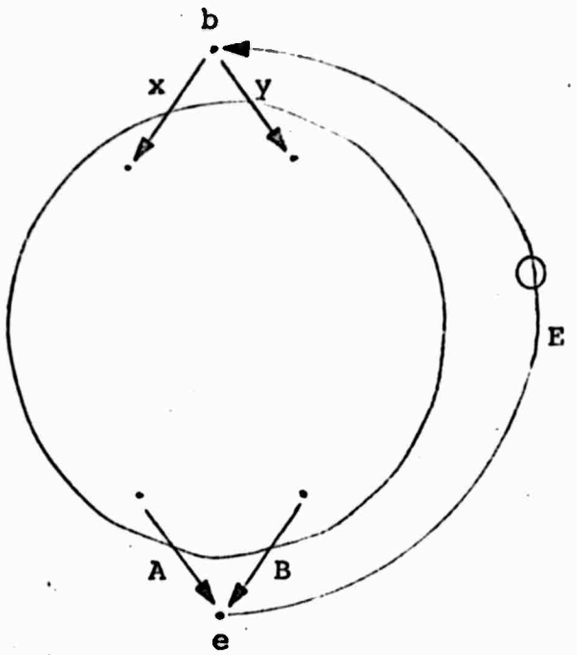
and by a modification like the one that leads from D4.1 to D4.2

.4



In D4.4 back flow has been represented by a single arc. Following the model of D4.4, we could also introduce back flow in C1.2 as follows:

D5.



This completion would correspond to the following set of semantic assumptions about the operation of the facility.

D6. .1 That the events which deliver the next value of x , the next value of y and the order for the next computation all coincide (b) ;

.2 That the events of receipt by the environment of the next A output value, the next B output value, and permission to generate a new order all coincide (e) ;

.3 That there is never more than one order, or one permission for an order (E) outstanding at any one time.

Now notice that D5 with the marked graph firing rule behaves exactly as C1.2 would under the special rule C3. Relative to rule C3, however, we have made substantial formal and semantic gains. First, initialization and termination have become event firings no different from any other in the schema; second, we have shown these rules to follow from a special set of semantic assumptions (D6.1 - .3) about the way in which the facility relates to its environment. Other assumptions lead to other back flows.

One can think of D5 as illustrating a standard paradigm for turning a primary flow diagram into a live and safe marked graph (see Chapter II, H4.1, .2 for definition). The paradigm is:

D7. .1 Identify all environment output collecting events to become a single event;

.2 Identify all environment input delivery events to become a single event;

.3 Add a back arc -- from collecting event to delivery event -- with a token.

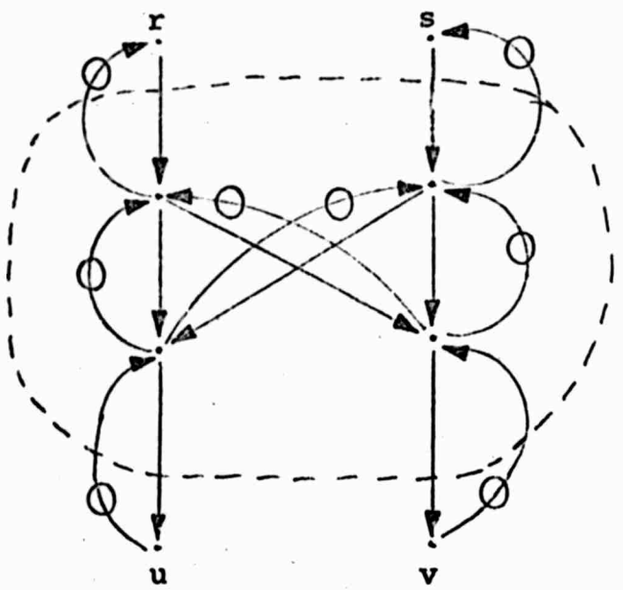
That a live and safe marked graph is produced in this way is proved in Chapter V.

In regard to the production facility, liveness means that the facility is so designed that it can continue to operate indefinitely. Safety means that there are never two concurrent holdings of the same condition or two

concurrent occurrences of the same event.[1] This implies for finite primary flow, that the facility can be operated with a finite set of processor storage units.

There is another standard paradigm for adding back flow to primary flow in order to produce a live and safe marked graph, illustrated by the next figure.

D8.



The rule is:

D9. To each primary flow arc, add a back arc with a token.

That this procedure produces live and safe marked graphs is proved in Chapter V.

One can express D9 as a policy in the design of a

---

[1]Safety also has something to do with the concept of "functionality" -- i.e. that successive outputs are functions of successive inputs and of nothing else -- but this topic is beyond the scope of this presentation.

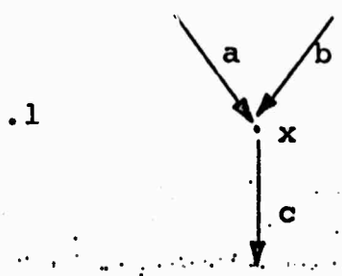
production facility, namely: at every point, give orders/ permissions for re-supply at the earliest time possible. This policy has the effect of maximizing concurrency in the operation of the facility and minimizing throughput rates. Other policies can be represented by yet other procedures for introducing back flow, but further discussion of this subject goes beyond the bounds of this presentation. Theorems pertaining to the evaluation of the degree of concurrency and to throughput rates are presented in Chapter V. Roughly, one can state: higher orders of concurrency represent greater demands on production resources, but tend to increase rates of throughput.

E. Internal Circuits

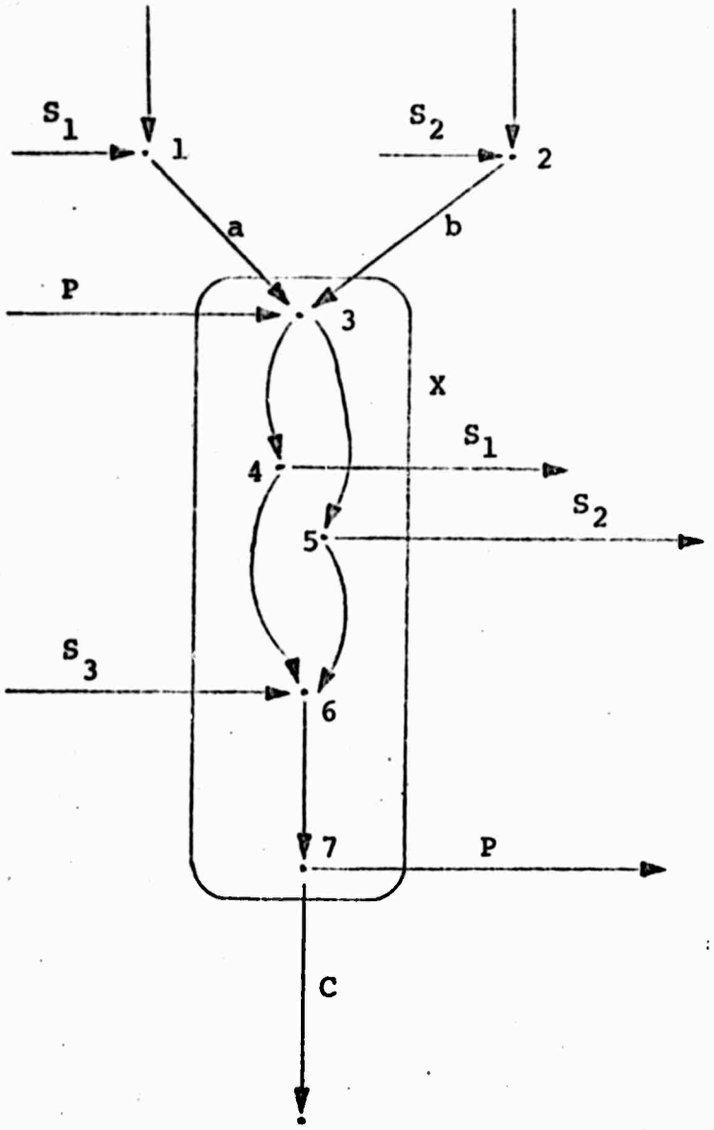
In addition to representing the flow of materials and orders through the facility one can also represent the flow of internal resources which may be reassigned to participate in various production events.

We may expand our picture of a single production event so as to show how reassignable resources participate in it. Example:

E1.



.2 (This is an expansion of Point 1.)



Event 1 shows an available storage unit S_1 being filled with a

Event 2, similarly for S_2 and b

Event 3 shows an available processor P beginning to perform X

Event 4 shows storage unit S_1 being released for reassignment

Event 5, similarly for S_2

Event 6 shows the commitment of a storage unit S_3 to the holding of the output C

Event 7 shows the processor P released for re-assignment

Now consider a facility in which there are N events, each requiring an identical reallocatable unit (e.g. N add events each requiring a given type and size of adder, or N storage events each requiring a given type and size of storage unit. One can examine the consequences of establishing a definite cyclic schedule of allocation for units of the required type.

E2.

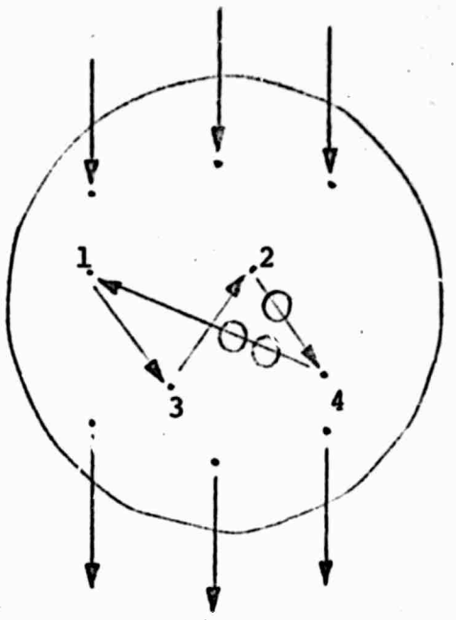


Figure E2 proposes an example where there are four such events, 1,2,3 and 4 and a particular cyclic schedule (one out of the six possible) for three identical resource units. Various questions now arise.

E3. .1 Is the new marked graph still live? If not, there may be some other way of distributing the three units on the circuit so as to yield a live graph. Or, one may discover four units are necessary.

.2 The token distribution on the circuit shown in E2 is not safe. It may be desired to find a safe distribution because it would remove the need for arbitration.

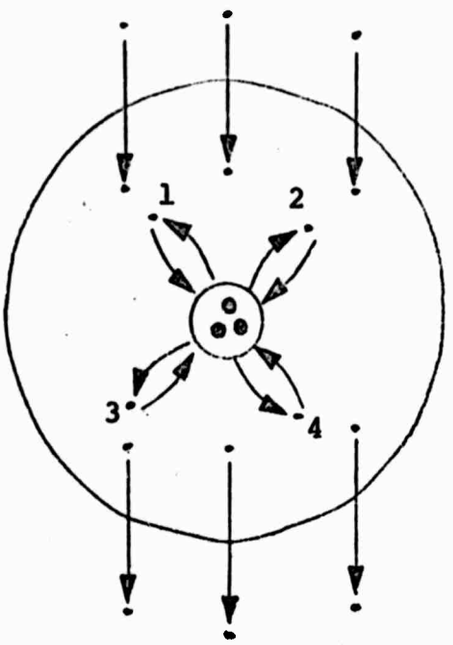
.3 What is the largest number of units which could be used on the circuit to advantage -- meaning that no larger number would increase the effective throughput rate and no smaller number would permit

as high a one.

Theorems are proved in Chapter V which provide algorithmic techniques for answering such questions.

While, with marked graphs one can explore the effect of various fixed cyclic schedules of allocation, one cannot represent and analyze the effect of resource pools. The next figure shows how, using a place connected to more transitions than is allowed in a marked graph, one can replace the circuit in E2 by a pool.

E4.



The effect of this arrangement is to insure that there will never be more than three concurrent firings among the transitions 1,2,3, and 4 .

There are other limitations on the descriptive power of marked graph models. Roughly speaking, one cannot model the effects of decisions with data dependent outcome, nor

can one conveniently model facilities in which each successive output requires the multiple traverse of various internal production cycles, some only a few times and others many times. These descriptive abilities also require the use of a more general class of Petri nets which we are not yet in a position to discuss systematically.

CHAPTER IV. PREVIEW OF MARKED GRAPHS

PREVIEW OF MARKED GRAPHS

The last chapter meant to give the reader some feel for how to apply meaning to marked graphs. This chapter means to give the reader some feel for how to apply mathematics to marked graphs. The results described and illustrated here are a significant sampling of the material in Chapter V where notations, proofs and algorithms are presented for the first time.

Many theorems in Chapter V concern the existence of certain markings or firing sequences in marked graphs under specified conditions. The proofs of these theorems provide efficient algorithms for determining whether or not the conditions hold, and constructing the marking on firing sequence if they do.

Liveness

The most important question about a production facility is: Will it operate? Will it function properly or are deadlocks present which may cause parts of it to halt?

In a marked graph model of a production facility, we define a vertex to be live if it may fire at some time. A vertex which is not live is said to be dead.

Our first theorem gives a necessary and sufficient condition for a vertex to be live.

Theorem 1: A vertex is live if and only if it is not contained in a blank circuit or a blank path from a blank circuit. (A set of arcs in a marked graph is said to be blank if it

contains no tokens.)

Example:

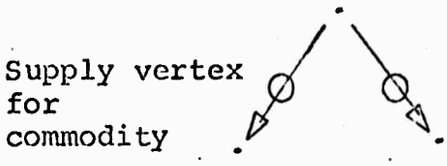
In this marked graph, the vertices a , b , and f are live. The vertices c and d are contained in a blank circuit and the vertex e is contained in a blank path from this circuit.

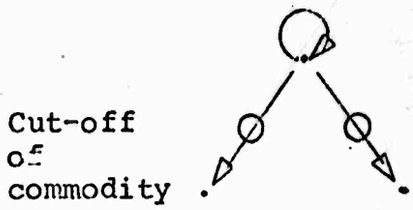
Even if a vertex is live it may only be able to fire a limited number of times. Vertex f above, for example, can fire only once.

Our next theorem gives a criterion for determining how many times a vertex v can fire starting from a given marking M . We call this number $D_M(v)$.

Theorem 2: In a marked graph with marking M , $D_M(v)$ is equal to the smallest number of tokens on a path from a dead vertex to v . (See example, Chapter V, page 91).

This theorem may be used to analyze the behavior of a production facility if some necessary commodity is cut off. We can represent this cut-off by placing a blank self loop on the vertex which represents the event which produces the commodity.





Theorem 2 then tells us how far the procedure will go without further supply of this item. It follows from Theorem 2 that if a marked graph contains no dead vertices every vertex may fire an unlimited number of times.

Such a marked graph is said to be <u>live</u>.

<u>Achievability Tests in Live Graphs</u>

In a live graph we may wish to know if a certain marking is achievable from a given marking M. Suppose we have a certain test marking T and we wish to know if M leads to a marking M' which contains T. Our next theorem provides a test to determine whether or not this is possible.

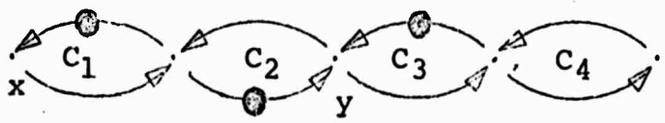
<u>Theorem 3</u>: In a live graph with marking M, let T be a test marking. M leads to a marking M' which contains T if and only if, for every circuit C, M places at least as many tokens on C as T places on C.

The proof of this theorem provides an algorithm which, if the circuit condition is satisfied, constructs a firing sequence which results in a marking M' which contains T.

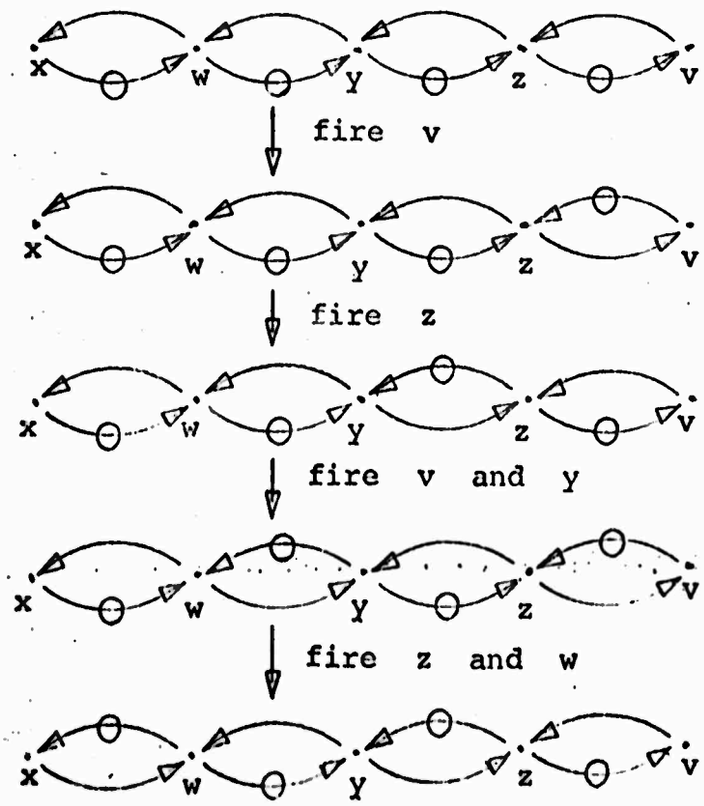
Example:

M: 

Can the vertices x and y fire concurrently? To find out, we use a test marking which makes both vertices firable.

T: 

The marking T places at most one token on each circuit (C_1, C_2, C_3 and C_4). The marking M places one token on each circuit. Thus M leads to a marking M' which contains T :

M: 

The next theorem allows us to determine whether one marking leads to another.

Theorem 4: In a live graph with marking M, M leads to M' if and only if M and M' place the same number of tokens on every circuit.

It may also be shown that:

Theorem 5: If M is live and M leads to M' then M' leads to M.

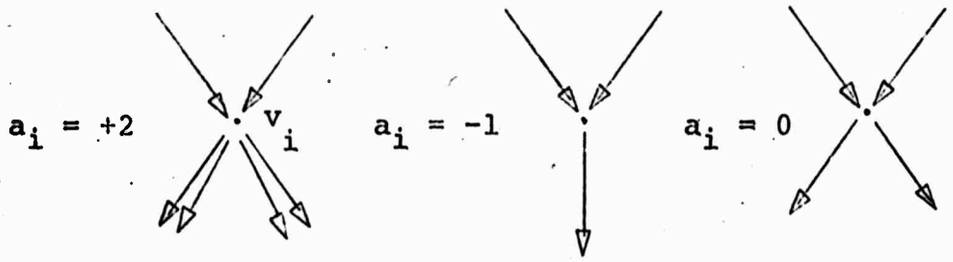
Chapter V, Section E contains theorems similar to Theorems 3 and 4 which do not assume either liveness or strong connectivity.

Maxima and Minima

Imagine a counter attached to each vertex in a marked graph which counts the number of times the vertex fires. We may wish to determine what the maximal or minimal value is of some function of these counter values. One interpretation of the counter values is that each vertex represents an event which produces a profit or loss to a business. In this case we may wish to determine what is the minimum amount of capital required to prevent the business from going into debt. (See example, Chapter V, Section G, page 127.) If the counter function is linear we have theorems which provide algorithms for determining what the optimal value of the function is and constructing an optimal sequence of vertex firings.

One interesting linear function of these counter values is the function $a_1x_1 + a_2x_2 \ldots + a_nx_n$ where the x_i is the value of the counter on the vertex v_i. The coefficient a_i is equal to the number of output arcs of v_i minus the number of input arcs of v_i.

Example:



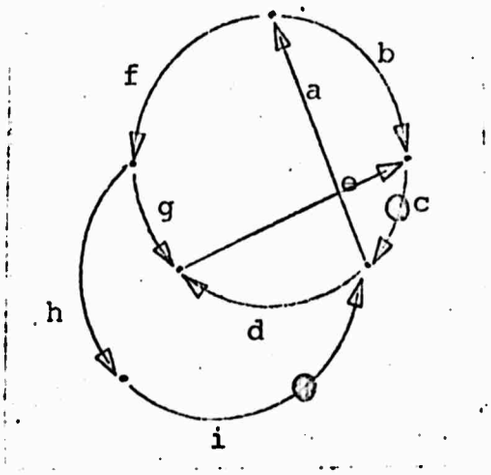
Clearly a_i is the net increase in marking size produced by a firing of v_i. This marking size function gives us a special case of the previously mentioned theorem. Thus we have a criterion for determining the number of tokens in "largest" marking M' to which a given marking M leads.

Theorem 6: Let g be a marked graph with marking M. The maximum number of tokens in a marking which M leads to is equal to the minimal number of tokens placed by M on a circuit which contains all the arcs of g.

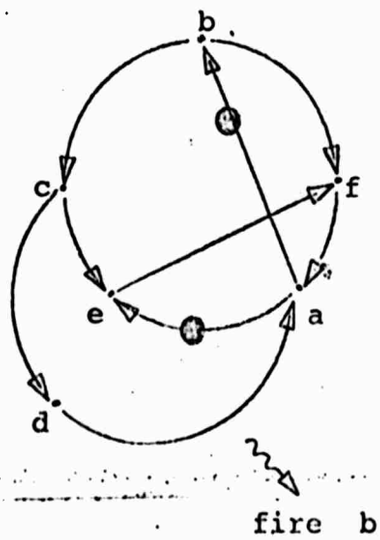
Example:

The largest marking which M leads to has size 4. A minimum circuit which contains every arc is $C = (abcdecafgecafhi)$. C passes through the arc c

three times and once through the arc i . Thus C has 4 tokens.

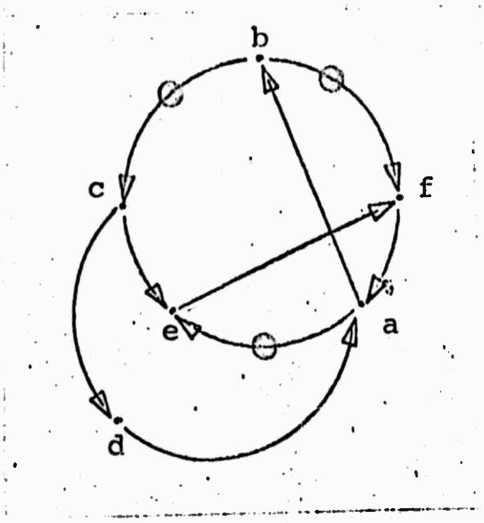


Here is a sequence of firings which brings us to the largest marking:



after firing a

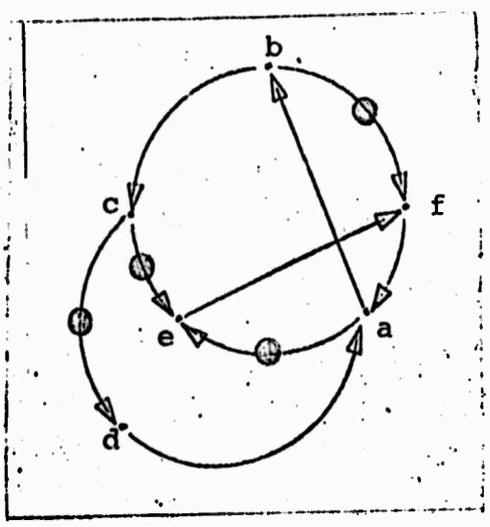
fire b



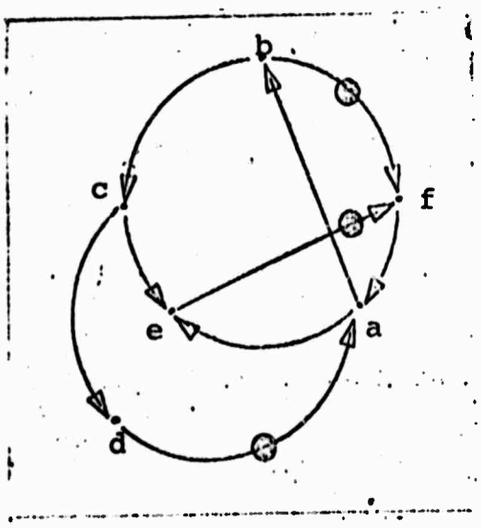
fire c .

This is a largest marking.

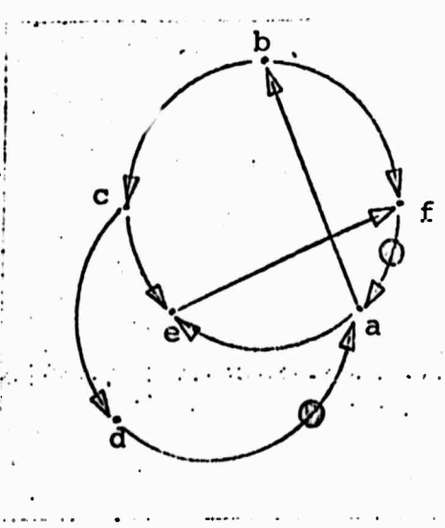
Now we return to the original marking.



fire d and e .



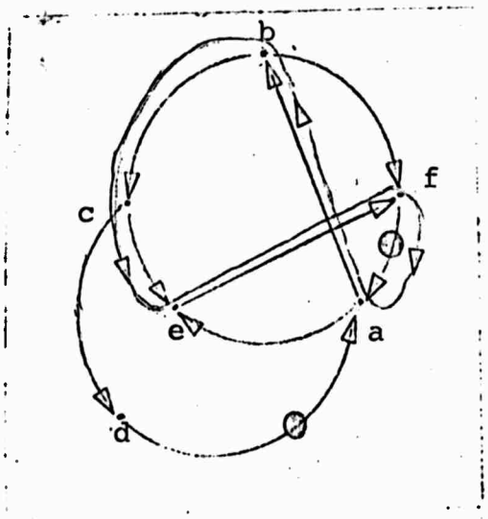
fire f



The size of the largest marking tells us how many processors would be required to perform a given cyclic task represented as a marked graph if no new timing restrictions are to result from the allocation of processors.

In the preceding example, one possible allocation is:

processor number	cyclic schedule of tasks
1	a,b,f
2	a,e,f
3	a,b,c,e,f
4	a,b,c,d



The route of processor 3 is shown in the graph.

Safety

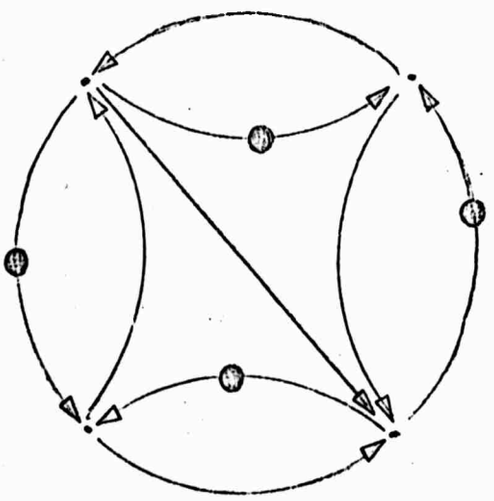
An arc in a marked graph is said to be safe if the maximum number of tokens which may ever appear on that arc is 1. Since a safe arc is either empty or has 1 token, it may be used

to represent a condition which either holds or does not hold, o a proposition which is either true or false.

Theorem: An arc in a live marked graph is safe if and only if it is contained in a circuit with 1 token.

Theorem 7: Every strongly connected graph has a live marking in which every arc is safe.

Example:
A live and safe marked graph:
For more examples see Chapter V, Section F.

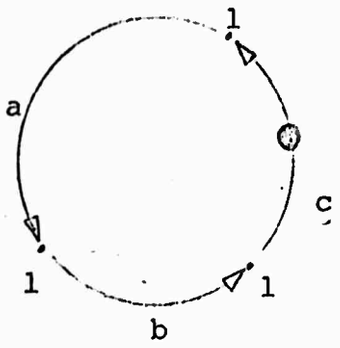


Throughput Rate

If each vertex in the graph is assigned a time duration, we may wish to determine what is the maximum average rate of firings per unit time for some vertex.

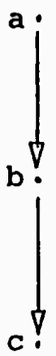
Theorem 8: The maximum average firing rate is the same for all vertices and is equal to the minimum ratio of the number of tokens on a simple circuit to the sum of the time delays of the vertices on the circuit.

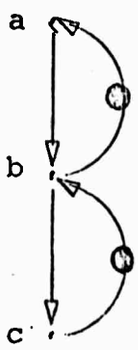
Example:



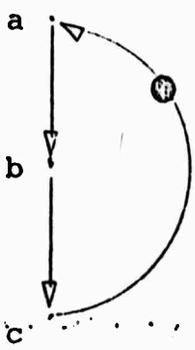
Each vertex in this graph is assigned a duration of 1. Each vertex can undergo one firing every three time units.

The circuit (a,b,c) has 1 token and 3 time units, giving a ratio of $\frac{1}{3}$. We may double this ratio by adding an extra token to the graph. (For more examples see Chapter V, Section H.) Here are three marked graph implementations of the task represented by the production scheme:

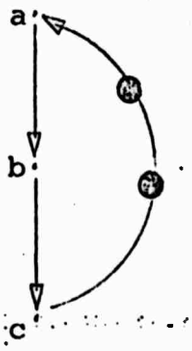




1.)



2.)



3.)

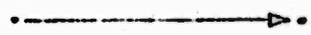
If each vertex firing requires 1 time unit, the average firing rates are:

1.) $\frac{1}{2}$ 2.) $\frac{1}{3}$ 3.) $\frac{2}{3}$.

MARKED GRAPHS MATHEMATICS

A. What is a Marked Graph?

A marked graph is a directed graph with a set of designated arcs. Arcs are designated by placing tokens on them.

undesignated: 

e.g.

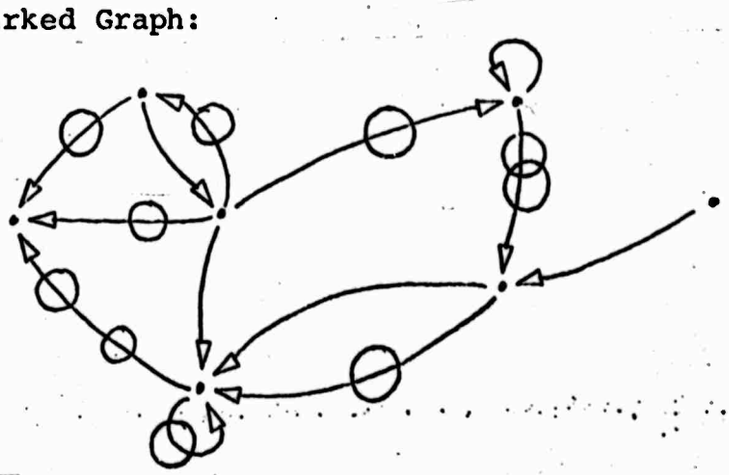
designated: 

Designated arcs may contain several tokens: 

One may think of the marking of a graph as the integer valued function which specifies, for each arc, the number of tokens placed upon it. (If no arc has more than one token, then this function is just the characteristic function* of the set of designated arcs.)

Example:

A Marked Graph:



---

*F is the characteristic function of S means:

$x \in S : \Rightarrow F(x) = 1$

$x \notin S : \Rightarrow F(x) = 0$

Formally we will define a marked graph mg thus:

$$mg \triangleq \langle\langle V, A, \dot{\dagger}, \dot{\dagger}\rangle, M\rangle$$

where 1. V is an at most denumerably infinite set of vertices.

2. A is an at most denumerably infinite set of arcs.

3. $\dot{\dagger}$ and $\dot{\dagger}$ are functions, from A to V. $\dot{\dagger}(a)$ is called the input vertex of a and $\dot{\dagger}(a)$ is called the output vertex of a. Also if $a \in \dot{\dagger}^{-1}(v)$ then a is called an output arc of v ; if $a \in \dot{\dagger}^{-1}(v)$, then a is an input arc of v .

A vertex may have only a finite number of input and output arcs.

4. M is a function from A to the non-negative integers. M is called the marking of mg.

5. $\langle V, A, \dot{\dagger}, \dot{\dagger}\rangle$ is the graph of the marked graph.

We will notationally treat $\dot{\dagger}$ and $\dot{\dagger}$ as relations, thus:

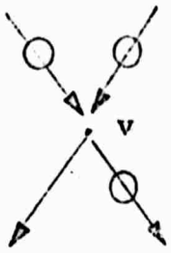
$\dot{\dagger}(a,x) \triangleq \dot{\dagger}(a) = x$; $\dot{\dagger}(a,-) \triangleq \dot{\dagger}(a)$ and $\dot{\dagger}(-,x) \triangleq \{a : \dot{\dagger}(a) = x\}$.

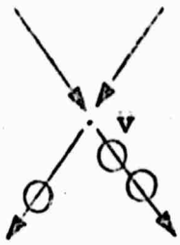
B. The Firing Operation

The firing operation is a type of transformation which takes one marking of a graph into another: Any vertex all of whose incoming arcs are marked may be fired by removing one token from all its input arcs and adding one token to all its output arcs.

E.g. Vertex v may be fired:

(1) before firing





(2) after firing

We say that a marking M leads to a marking M' , symbolically $M[\rightarrow M'$, if there exists a finite non-empty sequence of firings which transforms M , step-wise into M' . The definition of vertex firing makes clear that it is a reversible operation which we will call backward firing the vertex. We will write $M[=\rangle M'$ if $M[\rightarrow M'$ or $M' = M$.

The notation $M[x_1,x_2,x_3 \ldots x_n\rangle M'$ means that M can be transformed into M' by the sequence of vertex firings $x_1,x_2,x_3 \ldots x_n$. For a given finite sequence σ , of vertex firings we will also write $M[\sigma\rangle$ to mean the marking $M'|M[\sigma\rangle M'$, and $[\sigma\rangle M'$ to mean the $M|M[\sigma\rangle M'$.

Given a marked graph $\langle g,M\rangle$ we can define the strong reach of M denoted by $\vec{M}$ thus: $\vec{M} \triangleq \{M'|M[\rightarrow M'\}$; and the weak reach of M denoted by $\vec{\vec{M}}$ thus $\vec{\vec{M}} \triangleq \{M'|M[=\rangle M'\}$.

C. Paths and Tracks

A path is a sequence of arcs $a_1,a_2,a_3 \ldots a_n$ and a sequence of vertices $x_0,x_1,x_2 \ldots x_n$ such that the arc a_i connects the vertices x_{i-1} and x_i . We will denote paths by the following notations:

$\overline{\|a_1, a_2, a_3 \ldots a_n}$

or $\overline{\|x_0, x_1, x_2 \ldots x_n}$ which could designate any path on which the vertices appear in that order

or $\overline{P\|\sigma}$ where σ is a sequence of elements (arcs or vertices) and P is the name of the paths

A forwards directed path P is one in which each arc a_i connects x_{i-1} to x_i -- i.e. $\dot{\uparrow}(a_i, x_{i-1})$ and $\dot{\uparrow}(a_i, x_i)$ -- and can be denoted by $\overrightarrow{P\|x_0, x_1, x_2 \ldots x_n}$ or $\overrightarrow{P\|a_1, a_2 \ldots a_n}$.

A backwards directed path is one in which each arc a_i connects x_i to x_{i-1} and can be denoted by $\overleftarrow{P\|x_0, x_1 \ldots x_n}$ or $\overleftarrow{P\|a_1, a_2 \ldots a_n}$.

Given $\overrightarrow{P\|\sigma}$, we say:

If σ has an initial element x , then x is the initial element of P ;

If σ has a terminal element y , then y is the terminal element of P ;

If x and y exist, and $x \neq y$, then x and y are end points of P .

To say P begins at x or P ends at y is to imply that x and y are end points.

Any element of P which is not initial or terminal is called an inner element of P .

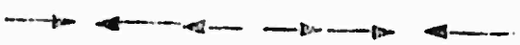
If P has no endpoints, it is called a track.

If P has no terminal endpoint, it is called a forward track.

If P has no initial endpoint, it is called a backward track.

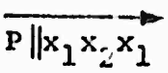
If the initial and terminal elements of P are the same, P is called a circuit.

Examples:



a path

a track



initial vertex of P : x_1
terminal vertex of P : x_1
an inner vertex of P : x_2
endpoints of P : none

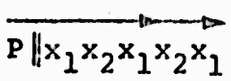
P is a circuit

P is a track

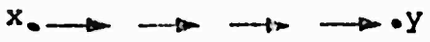
(P is also a forward track and a backward track)

a forward track

a backward track



a track
P does not begin at x_1



This path begins at x and ends at y .
Not a forward track, not a backward track, not a track.

D. Lengths of Paths and Sequences

Given a structure S with elements e and a property P we define:

$\|S|P\|$ $\triangleq$ The number of elements of S which have property P

Examples:

Assume $P\|\langle a_i\rangle_1 \leq i \leq n$ is a directed path in $\mathfrak{M} = \langle\langle V, A, \dot{\uparrow}, \dot{\uparrow}\rangle, M\rangle$

Then $|P|A|$ — the arc length of P, is the number of arcs in P.

$|P|M|$ — the token length of P, is the number of tokens on P: more exactly:

$$|P|M| = \sum_{i=1}^{n} M(a_i) .$$

If $x \in V$

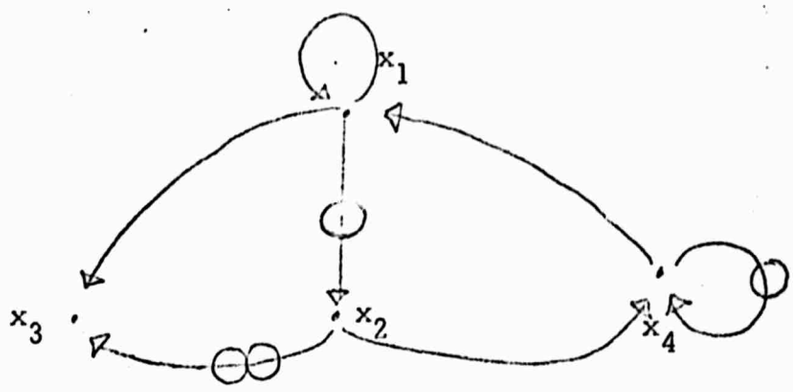
then $\|P|x\|$ — the number of times that the vertex x appears as a vertex of the path.

$\|P|\dot{\uparrow}(-,x)\|$ — the number of arcs of P which are output arcs of x.

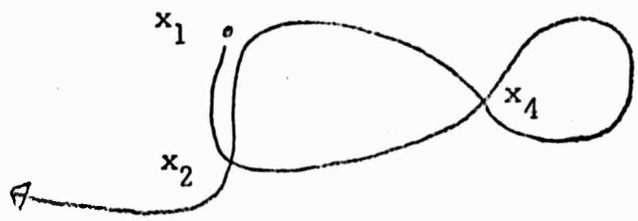
We write {P} to mean the subgraph covered by the path P.

$\|\{P\}|M\|$ — the reduced token length of P is the number of tokens on the subgraph which P covers. More exactly:

$$\|\{P\}|M\| = \sum_{a \in \{P\}} M(a) .$$



The graph



The path P

$$\| P \mid A \| = 6$$

$$\| P \mid M \| = 5$$

$$\| P \mid \{x_2, x_1\} \| = 4$$

$$\| P \mid \dot{\iota}(-x_1) \| = 2$$

$$\| \{P\} \mid M \| = 4$$

$$\| \{P\} \mid \dot{\iota}(-, x_2) \| = 1$$

The same notations can be used for arbitrary sequences, whether they are paths or not. For example, if σ is a sequence of vertices $\langle x_3, x_4, x_1, x_3, x_2 \rangle$ then $\| \sigma \mid x_3 \| = 2$. Define

$$\| S \mid f(P_1, P_2, P_3 \ldots P_n) \| \triangleq f(\| S \mid P_1 \|, \| S \mid P_2 \|, \| S \mid P_3 \| \ldots \| S \mid P_n \|)$$

Example:

$$\| P \mid x_1 - x_3 \| = \| P \mid x_1 \| - \| P \mid x_3 \| = 2 - 1 = 1$$

Theorem D1

Let $M[x\rangle M'$ and let P be a forward directed path. Then $\| P|M \| = \| P|M' \|$ if x is not an endpoint of P. If P begins at x then $\| P|M' \| = \| P|M \| + 1$; if P ends at x then $\| P|M' \| = \| P|M \| - 1$.

Proof

→ Suppose x is not an endpoint of P. Then $\| P \mid \dot{\uparrow}(-,x) \| = \| P \mid \dot{\uparrow}(-,x) \| = \| P \mid x \|$. Only arcs which are inputs or outputs of x have a different value in M' than in M: each input arc is decreased by 1, each output arc increased by 1. It follows that, if $k = \| P \mid x \|$, then the $\| P \mid M' \| = \| P \mid M \| - k + k$.

← Suppose P ends at x. Then, by what was just shown $\| P \mid \dot{\uparrow}(-,x) \| = \| P \mid \dot{\uparrow}(-,x) \| + 1 = \| P \mid x \| = k$.
Therefore $\| P \mid M' \| = \| P \mid M \| - k + (k-1) = \| P \mid M \| - 1$.
Similarly if P begins at x . Q.E.D.

Immediate consequences:

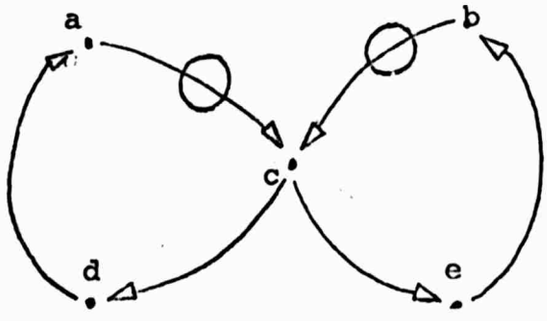
D2. Assume $M[x\rangle M'$; if P is a track then $\| P \mid M' \| = \| P \mid M \|$; if P is a forward track with a beginning at x then $\| P \mid M' \| = \| P \mid M \| + 1$; if P is a backward track beginning at x beginning then $\| P \mid M' \| = \| P \mid M \| - 1$.

D3. Assume that $M[\sigma\rangle M'$ and that, in σ, the vertices x and y were fired an equal number of times. Then for all paths P from x to y, $\| P \mid M \| = \| P \mid M' \|$. Thus the pair of markings, M and M' determine a partition of the vertices of the graph

into equivalence classes via the relation: x and y are fired an equal number of times in σ . We will call this the vertex partition of (M,M') .

D4. Given $M[\sigma\rangle M'$ and that, for vertices x and y, $\| \sigma \mid x \| - \| \sigma \mid y \| = n$. Then, for all paths P from x to y, $\| P \mid M' \| = \| P \mid M \| + n$. In particular, if, for all arcs a $\| \sigma \mid \dot{\tau}(a,-) \|$ and $\| \sigma \mid \dot{\tau}(a,-) \|$ are known then $M'(a) - M(a)$ is known. Now letting ζ be a variable which ranges over all vertices of the graph we may think of σ as defining a function $\| \sigma \mid \zeta \|$, mapping each vertex x to the integer which specifies the number of times x was fired in σ . We have shown that the function $\| \sigma \mid \zeta \|$ exactly specifies M', if M was given. Thus, if $M[\sigma_1\rangle M_1$ and $M[\sigma_2\rangle M_2$ and $\| \sigma_1 \mid \zeta \| = \| \sigma_2 \mid \zeta \|$ then $M_1 = M_2$.

Example:



Marking M

$M \ [\sigma\rangle \ M$

σ may be any of the following sequences

c, e, b, d, a

c, e, d, a, b

c, e, d, b, a

c, d, a, e, b

c, d, e, b, a

c, d, e, a, b

The following is a natural generalization of D1.

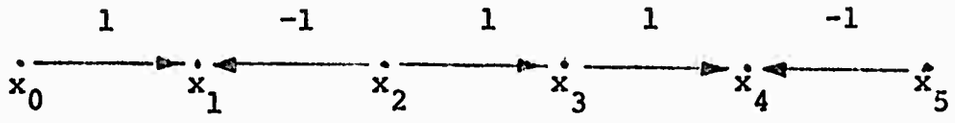
Let P be the undirected path $x_0, a_0, x_1, a_1, x_2, a_2 \ldots x_n$

Define $\| P|M \| = \sum_{i=0}^{n-1} \delta_i M(a_i)$

where $\delta_i = 1$ if $\dot{\uparrow}(a_i, x_i)$

and $\delta_i = -1$ if $\dot{\uparrow}(a_i, x_i)$

Example



Theorem D5

Let $M[x\rangle M'$ and let P be a path. Then $\| P|M \| = \| P|M' \|$ if x is not an endpoint of P. If P begins at x then $\| P|M' \| = \| P|M \| + 1$; if P ends at x then $\| P|M' \| = \| P|M \| - 1$.

Proof

Suppose x is not an end point of P. If it is not a vertex of P at all then neither do any of its input or output arcs lie on P, and the firing of x cannot change the token length of P. Now suppose x occurs one or more times on the path P. We can assign to each occurrence of x in P two arcs of P:
If $x = x_i$ $0<i<n$ then assign the arc pair $\langle a_i, a_{i+1} \rangle$
if $x = x_n = x_0$ then assign the arc pair $\langle a_0, a_n \rangle$.
For the input and output arcs a of x, define $j(a) = +1$ if a is an input of x and $j(a) = -1$ if a is an output of x. Then, if $\langle a, a' \rangle$ is an arc pair of x on the path P

$j(a) = \delta(a)$ and $j(a') = -\delta(a)$. Thus these two arcs contribute $c = j(a)M(a) - j(a')M(a')$ to $\|P|M\|$. By firing the vertex, the number of tokens on an input or output arc a of x is changed by $-j(a)$. Therefore the arc pair $\langle a,a' \rangle$, after firing, contributes $c' = j(a)[M(a)-j(a)] - j(a')[M(a')-j(a')]$ to $\|P|M'\|$ but $c' = j(a)M(a) - j(a)^2 - j(a')M(a') + j(a')^2$

$$c' = c + j(a')^2 - j(a)^2 = c .$$

If P ends at x then, in addition to a set of arc pairs which P might contain because of other occurrences of x on P , the firing of x also affects a_n (but not a_0). The contribution of a_n to $\|P|M\|$ is $c = j(a_n)M(a_n)$; after firing the contribution is $c' = j(a_n)[M(a_n) - j(a_n)] = c - 1$. Hence $\|P|M'\| = \|P|M\| - 1$; similarly, if P begins with x .

Restrictions

Let G be a graph with marking M . Let G' be any subgraph of G .

Definition

$M|G'$ Δ The restriction of the function M to G'

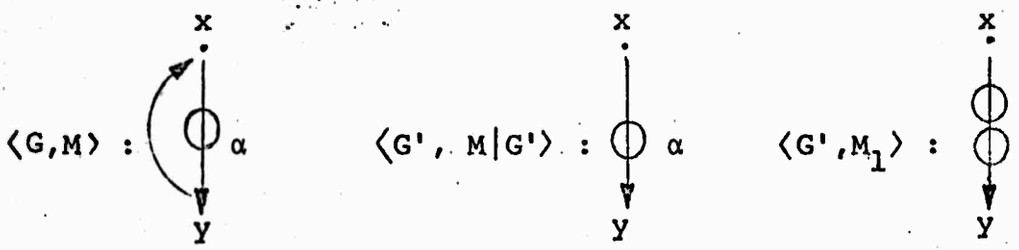
$\vec{M}|G'$ Δ The restriction of the weak reach of M to G'

This is the class of markings M_1 such that there exists M_2 such that $M_2 \in \vec{M}$ and $M_1 = M_2|G'$

$\overrightarrow{M|G'}$ Δ The weak reach of $M|G'$.

In general $\overrightarrow{M|G'} \neq \vec{M}|G'$.

Example:



Vertex x is not firable in G' but not in G .
Thus the marking M_1 is in $\overrightarrow{M|G'}$ but not in $\vec{M}|G'$.

Theorem D6

In a marked graph G with marking M where $G' \subseteq G$, $\vec{M}|G' \subseteq \overrightarrow{M|G'}$.

Proof:

We will show that $(M_1 \in \vec{M}) \Longrightarrow (M_1|G' \in \overrightarrow{M|G'})$.
Let $M_1 \in \vec{M}$. If $M_1 = M$ then $M_1|G' = M|G' \in \overrightarrow{M|G'}$.
Now assume $M_1 \neq M$. In that case, $M[\rightarrow M_1$. We will now prive that for any vertex x ,

$$(M[x\rangle M_1) \Longrightarrow (M_1|G' \in \overrightarrow{M|G'})$$

Suppose $x \in G'$. If x is firable in M then all its input arcs must contain tokens. Thus all its input arcs in G' must contain tokens. Thus x is firable in M|G' and $M|G'[x\rangle$ exists. Clearly $M[x\rangle|G' = M|G'[x\rangle$. Thus $M_1|G' \in \overrightarrow{M|G'}$. Now assume $x \notin G'$ then the token content of every arc in G' is unchanged by the firing of x . Thus $M_1|G' = M|G'$ and $M_1 \in \overrightarrow{M|G'}$.

Now assume σ is any sequence such that $M[\sigma\rangle$ exists. Let $\sigma = s_1, s_2, s_3 \ldots s_n$ and $M[s_1\rangle M_1[s_2\rangle M_2 \ldots [s_n\rangle M_n$. Assume $M_i|G' \in \overrightarrow{M|G'}$. Then $M_i[s_{i+1}\rangle M_{i+1}$ and by the above argument $M_{i+1} \in \overrightarrow{M_i|G'}$. However since $M_i|G' \in \overrightarrow{M|G'}$ then $\overrightarrow{M_i|G'} \subseteq \overrightarrow{M|G'}$ thus $M_{j+1} \in \overrightarrow{M|G'}$. By induction $M_n|G' \in \overrightarrow{M|G'}$.

Q.E.D.

E. Liveness

Definition:

E1. A vertex x in $\mathcal{G} = \langle g, M\rangle$ is live if there exists a firing sequence σ which contains x. More exactly: $\exists\sigma|M[\sigma\rangle$ exists and $\|\sigma|x\| \geq 1$.

E2. A marked graph is live if all of its vertices are live.

In this section we will discuss criteria by which one can determine whether a vertex or a graph is live. We will also discuss the connection between liveness as a "static" fact about a given marked graph $\mathcal{G}$, and the behavior of $\mathcal{G}$ when it is transformed by firing sequences.

Lemma E3.

Let g be a directed graph in which every vertex has, at most, a finite number of input arcs. Assume that g contains a vertex x_0 which can be reached from an infinite number of vertices y in g -- i.e. $(\forall y)\ (\exists P)\ (\overleftarrow{\|x_0 \ldots y})$. Then there exists an infinite

path $\overleftarrow{P' \| x_0 \ldots}$.

Preliminary Definition.

For a vertex x, define the backward reach of x denoted $\overleftarrow{x}$ as $\{y: (\exists P)(\overleftarrow{P \| x \ldots y})\}$.

Proof

We shall inductively construct an infinite sequence of vertices and arcs $x_0, a_0, x_1, a_1, x_2, a_2 \ldots$ |

.1 $(\forall a_i:$ $x_i \overleftarrow{\quad a_i \quad} x_{i+1}$

.2 $\overleftarrow{x_i}$ is infinite

Note first that, by the hypothesis of the theorem $\overleftarrow{x_0}$ is infinite. Now examine the set of all vertices z_i from which x_0 can be reached by a path with a single arc. There must be such vertices, or the backward reach of x_0 cannot be infinite. At least one of these vertices, say z_K, has infinite backward reach. This is because $\overleftarrow{x_0} = \{z_i\}_i \cup \bigcup_i \overleftarrow{z_i}$ and the union of a finite number of finite sets is at most finite, contradicting the assumption that the backward reach of x_0 is infinite. Pick an arc $x_0 \overleftarrow{\quad a_1 \quad} z_K$ and let x_1 be z_K. Now given that $x_0, a_0, x_1, a_1 \ldots x_n$ has been constructed we can construct a_n and x_{n+1} by the same argument.

Theorem E4

A vertex x in $\langle g,M \rangle$ is live if and only if for all infinite paths $\overleftarrow{P}\|x\ldots$, $\|P|M\| \geq 1$.

Proof:

→ Suppose there exists a $\overleftarrow{P}\|x\ldots \mid \|P|M\| = 0$. By D1 we know that for arbitrary firing sequences $M[\sigma\rangle M'$ $\|P|M'\| = 0$. In particular no firing sequence results in a marking which places one or more tokens on the first arc of P . But that arc is an input of x . Therefore M does not lead to a marking in which x is firable.

← Suppose that, for all infinite paths $\overleftarrow{P}\|x\ldots$, $\|P|M\| \geq 1$. Let B_x be the subgraph of g obtained by taking the union of all backward directed paths P from $x \mid \|P|M\| = 0$. If there are more, let B_x consist of the vertex x alone. B_x contains at most a finite number of vertices. If it contained an infinite number then, by lemma E3, B_x would contain an infinite path $\overleftarrow{P'}\|x\ldots$ with $\|P'|M\| = 0$, contradicting the hypothesis.

Next observe that B_x must contain at least one firable vertex. First, if B_x consists of the vertex x alone, then by construction, x must be firable. Second, note that B_x must be circuit free, for otherwise an infinite path $\overleftarrow{P'}\|x\ldots$ can be constructed

in B_x . Since B_x is finite and circuit free it must contain finite paths of maximal length $P\overleftarrow{|x \ldots y}$. But by construction, y must be firable, the blank path P can be extended by adding an empty input arc of y .

Let y be a firable vertex of B_x , $y \neq x$. Assume $M[y\rangle M'$ and define B'_x relative to M' just as B_x was defined relative to M . We shall show that B'_x is properly contained in B_x .

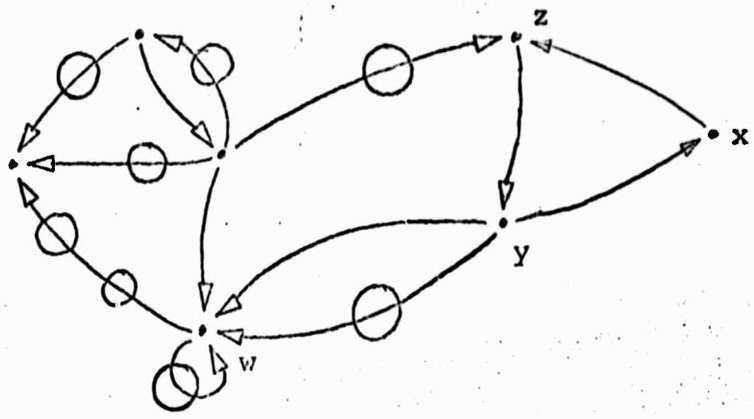
(1) $y \in B_x$ and $y \notin B'_x$ because by theorem D1 the token length of every path from y to x must have increased by 1 .

(2) If $z \in B'_x$ and $z \notin B_x$ then the token length of some path from z to x must have decreased to 0 as a result of firing vertex y . By D1 this shows that $z = y$.

Thus we have shown that, starting with M , one can construct a finite firing sequence which terminates with the firing of vertex x . Q.E.D.

Example:

The vertices w,x,y, and z are dead and contained in the blank backwards track $\overleftarrow{|w,y,z,x,y,z,x\ldots}$.
All other vertices are live.



Theorem E5

A vertex firing in a marked graph may only affect the liveness of the fired vertex.

Proof:

Let $M[x\rangle M'$. By E1, a vertex is v dead if and only if there exists a backwards blank track T which contains v . Assume $|T|M|| > 0$ and $|T|M'|| = 0$. By D1 only paths which end at the vertex x may decrease in token length as a result of the firing. Thus $v = x$.

Definition:

1. Let x be a vertex in $g_M = \langle g, M\rangle$. If g_M contains any dead vertices d and any paths P of the form $\overleftarrow{P}\|x \ldots d$ then define

$$D_M(x) = \min_{d,P} |\overleftarrow{P}\|x \ldots d|M\| .$$

If there are no such paths then $D_M(x)$ is undefined.

2. A firing sequence σ kills the vertex x if $\|\sigma|x\| \geq 1$ and x is dead in $M[\sigma\rangle$.

Theorem E6

Let x_0 be a vertex in $\mathcal{G} = \langle g, M\rangle$. Then:

(1) If there exists a sequence σ which kills x_0 and $\|\sigma|x_0\| = N$ then $D_M(x_0) = N$.

(2) If $D_M(x_0) = N$ then for all σ which kill x_0, $\|\sigma|x_0\| = N$.

(3) If $D_M(x_0) = N$ there exists a firing sequence which kills x_0.

Proof by induction on N.

N=1

(1) Suppose there exists a sequence $M[\sigma\rangle M'$ which kills x_0 and $\|\sigma|x_0\| = 1$. Since x_0 is dead in $\mathcal{G}' = \langle g, M\rangle$, by E4 there must exist a backward track $\overleftarrow{T\|x_0, x_1, x_2, x_3 \ldots}\,|\,\|T|M'\| = 0$. Let x_i be the last vertex of T such that $\|\sigma|x_i\| \geq 1$. Then clearly x_{i+1} must be dead in $\mathcal{G}$ and by Theorem D1 we must have $\|\overleftarrow{T x_0, x_1 \ldots x_{i+1}}|M\| = 1$. On the other hand, no dead vertex in $\mathcal{G}$ could have yielded a shorter token path to x_0 for otherwise, by E5, x_0 would be dead relative to M. Hence $D_M(x) = 1$.

(2) $D_M(x_0) = 1$. By E5 we know that x_0 is live. Therefore there exists a firing sequence which fires x_0. By D1 such a firing sequence can fire x_0 at most once,

after which x_0 is dead.

(3) By D1, no sequence which kills x_0 can contain more than 1 firing of x_0. By E4, none can contain less.

N ⇒ N+1

Assume the theorem for $N \geq 1$

(1) Assume there exists σ killing x_0 such that $\|\sigma|x_0\| = N + 1$. Let $\sigma = s_1, s_2, s_3 \ldots s_k$. Let s_i be the first occurrence of x_0. Let $\sigma_1 = s_1, s_2, s_3 \ldots s_i$ $\sigma_2 = s_{i+1}, s_{i+2} \ldots s_k$ and $M[\sigma_1\rangle M_1$. Now by assumption $\|\sigma_2|x_0\| = N$. Thus $D_{M'}(x_0) = N$. We will show that there exists a dead vertex d' relative to M_1 such that $\|\sigma_1|d'\| = 0$ and $\exists \overleftarrow{P|x \ldots d'} \mid \|P|M_1\| = N$.

Let d be any dead vertex relative to M_1 such that there exists $\overleftarrow{P\|x \ldots d}$ such that $\|P|M_1\| = N$.

Now by E4 there exists a backwards blank track T containing d. If the track contains a blank circuit, every vertex on the circuit was dead relative to M. Choose any vertex on the circuit as d'. If the track does not contain a circuit, it contains an infinite number of vertices, not all of which could have been included in σ_1. Then choose d' such that $d' \notin \sigma_1$.

In either case $\|T|M_1\| = 0$. Since T includes d' and d there exists a path $\overleftarrow{P_1\|x \ldots d'}$ such that

$\|P_1|M_1\| = N$. Since $\|\sigma_1|x_0\| = 1$ and $\|\sigma_1|d'\| = 0$ $\|P_1|M\| = N + 1$. Thus $D_M(x_0) \leq N + 1$. However if $D_M(x_0) < N + 1$ then by assumption $\|\sigma|x_0\| < N + 1$ thus, $D_M(x_0) = N + 1$.

(2) Assume $D_M(x_0) = N + 1$. By D1, for all σ , if σ kills x_0 then $\|\sigma|x_0\| \leq N + 1$. If there exists a sequence σ such that $\|\sigma|x_0\| = K < N + 1$ by inductive assumption (2) $D_M(x_0) = K$ contradicting assumption. Thus for all σ killing x_0 $\|\sigma|x_0\| = N + 1$.

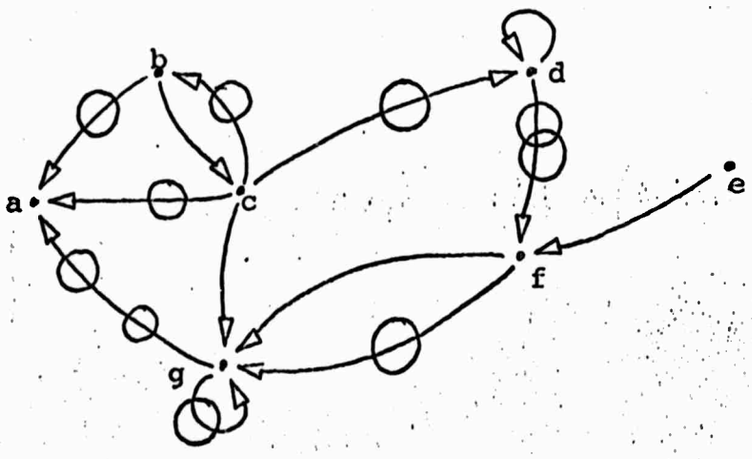
(3) Since we know that x_0 is live and since it surely cannot be fired more than $D_M(x_0)$ times, a sequence which kills x_0 surely must exist.

Q.E.D.

Corollary E6a

If $M[x\rangle M'$ then $D_{M'}(x) = D_M(x) - 1$.

Example:



$D_M(a) = 4$
$D_M(b)$ is undefined
$D_M(c)$ is undefined
$D_M(d) = 0$
$D_M(e)$ is undefined
$D_M(f) = 2$
$D_M(g) = 2$

The reader is encouraged to verify this data by placing $M(\alpha)$ paper tokens on each arc α and transforming the marking by the firing rule. Note that the graph may be made live by adding a token to the blank self loop on d or by removing the self loop.

Composition of Marked Graphs

Define $\mathcal{G}_1 \oplus \mathcal{G}_2 = \langle g', M' \rangle$ where

$g' = g_1 \oplus g_2 \triangleq \langle V_1 \cup V_2, \Lambda_1 \cup \Lambda_2, \dagger_1 \cup \dagger_2, \dot{\dagger}_1 \cup \dot{\dagger}_2 \rangle$ and

$M' \triangleq$ For all arcs a:

(1) If $a \in \Lambda_1$ and $a \notin \Lambda_2$ then $M'(a) = M_1(a)$

(2) If $a \notin \Lambda_1$ and $a \in \Lambda_2$ then $M'(a) = M_2(a)$

(3) If $a \in \Lambda_1$ and $a \in \Lambda_2$ then $M'(a) = M_1(a) + M_2(a)$

Killing Vertices

Let $\mathcal{G} = \langle g, M \rangle$ and let X be a set of vertices.

$X = \{x_1, x_2, x_3 \ldots x_n\} \subseteq g$

Define $\mathcal{G}_X = \mathcal{G} \oplus_{x_1} \oplus_{x_2} \oplus_{x_3} \ldots \oplus_{x_n}$

Clearly all $x \in X$ are dead in $\mathcal{G}_X$.

We will call the transformation from $\mathcal{G}$ to $\mathcal{G}_X$ killing the set X of vertices.

Now let y be a single vertex,

Define $\mathcal{G}_y = \mathcal{G}_{\{y\}}$

Now let $\mathcal{G} = \langle g, M \rangle$ and let $\mathcal{G}_y = \langle g_y, M_y \rangle$.

Let x be a vertex in $\mathcal{G}$,

Define $D_M(x|y) = D_{M_y}(x)$

Theorem E6b

$$D_M(x|y) = \max_{M[\sigma\rangle M'} \left(\|\sigma|x\| - \|\sigma|y\| \right)$$

Proof:

Assume $D_M(x|y) = D_M(x)$ then for all σ

$\|\sigma|x\| \leq D_M(x|y)$. Since $\|\sigma|y\| \geq 0$,

$\|\sigma|x\| - \|\sigma|y\| \leq D_M(x|y)$.

Now assume $D_M(x|y) = N \neq D_M(x)$ then there must exist a path $\overrightarrow{P}\|x...y$ such that $\|P|M\| = N$. If there exists a sequence σ such that $M[\sigma\rangle M'$ and $\|\sigma|x\| - \|\sigma|y\| = S > N$ then $N - S < 0$ and $\|P|M'\| = \|P|M\| - \left(\|\sigma|x\| - \|\sigma|y\|\right)$, $\|P|M'\| = N - S < 0$. But the size of any path is non-negative, thus $\|\sigma|x\| - \|\sigma|y\| \leq N = D_M(x|y)$. We will now show there exists σ_1 such that $M[\sigma_1\rangle$ exists and $\|\sigma|x\| - \|\sigma|y\| = D_M(x|y)$. Recall that $D_M(x|y) = D_{M_y}(x)$ where M_y is the marking of the graph $\mathcal{G}$ with y killed. By E6, there exists a firing sequence σ_1 in $\mathcal{G}_y$ such that $\|\sigma_1|x\| = D_{M_y}(x) = D_M(x|y)$. Since y is dead, $\|\sigma_1|y\| = 0$. By D7 if $M_y[\sigma_1\rangle$ exists then $M[\sigma_1\rangle$ exists. Thus there exists σ_1 such that $M[\sigma_1\rangle$ exists, $D_M(x|y) = \|\sigma_1|x\|$, $\|\sigma_1|y\| = 0$, and hence $\|\sigma_1|x\| - \|\sigma_1|y\| = D_M(x|y)$. Thus $\max_{M[\sigma\rangle M'} (\|\sigma|x\| - \|\sigma|y\|) = D_M(x|y)$.

Corollary

Let $M[x\rangle M'$. For all $y \neq x$, $D_{M'}(x|y) = D_M(x|y) - 1$.

Corollary

Let $M[x\rangle M'$. For all $y \neq x$,

$D_{M'}(y|x) = D_M(y|x) + 1$ if and only if

$D_M(y|x) \neq D_M(y)$

Proof:

→ If $D_M(y|x) = D_M(y)$ then

$D_{M'}(y) \leq D_M(y) = D_M(y|x)$. However,

$D_{M'}(y|x) \leq D_{M'}(y)$ thus $D_{M'}(y|x) \leq D_M(x|x)$.

← If $D_M(y|x) \neq D_M(y)$ then $D_M(y|x) < D_M(x)$ and there exists a path $\overleftarrow{P\|y...x}$ such that $\|P|M\| = D_M(y|x)$. Furthermore since $D_{M'}(y|x) = \|P|M'\| = \|P|M\| + 1 = D_M(y|x) + 1$.

Q.E.D.

Theorem E6c

Let $G' \subseteq G$

$\overleftarrow{M}|G' = \overleftrightarrows{M|G'}$ if and only if for all vertices $x,y, \in G'$, $D_M(y) = D_{M|G'}(y)$ and $D_M(x|y) = D_{M|G'}(x|y)$.

Proof:

→ Assume $D_M(y) \neq D_{M|G'}(y)$ or $D_M(x|y) \neq D_{M|G'}(x|y)$. Clearly $D_M(y) \leq D_{M|G'}(y)$ and $D_M(x|y) \leq D_{M|G'}(x|y)$, thus $D_M(y) < D_{M|G'}(y)$ or $D_M(x|y) < D_{M|G'}(x|y)$. In either case by E6 and E6b there exists a sequence σ such that $M|G'[\sigma\rangle$ exists but $M[\sigma\rangle$ does not exist. Thus there does not exist M' such that $M[\rightarrow M'$ and $M'|G' = M|G'[\sigma\rangle$. Hence $\overleftarrow{M}|G' \neq \overleftrightarrows{M|G'}$.

← Assume $\forall x,y, \in G'$

(1) $D_M(y) = D_{M|G'}(y)$ and $D_M(x|y) = D_{M|G'}(x|y)$. By D6,

$\vec{M}|G' \subseteq \vec{\vec{M}}|G'$. We will show that under assumption (1) $M_1 \in \vec{\vec{M}}|G'$ implies $M_1 \in \vec{M}|G'$, that is, there exists M_2 such that $M[\rightarrow M_2$ and $M_2|G' = M_1$. Assume $M_1 = M|G'$ then $M_1 \in \vec{\vec{M}}|G'$ and $M_1 \in \vec{M}|G'$. Now assume $M_1 \neq M|G'$. Then $M|G'[\rightarrow M_1$. We will now prove that for any single vertex firing x such that $M|G'[x\rangle M_1$, there exists M_2 such that $M[\rightarrow M_2$ and $M_2|G' = M_1$, and condition (1) is preserved by the firing.

Let $x \in G'$ and $M|G'[x\rangle M$. Now assume there does not exist a firing sequence σ such that $M[\sigma\rangle$ exists, $\|\sigma|x\| = 1$, and for all $y \in G'$ $y \neq x$, $\|\sigma|y\| = 0$. Then $D_M(x|y) = 0$. However $D_{M|G'}(x|y) \geq 1$ since x is firable. Thus $D_{M|G'}(x|y) \neq D_M(x|y)$, contradicting assumption (1). Hence the sequence σ must exist. Clearly $M[\sigma\rangle|G' = M_1$. Now let $M[\sigma\rangle M_2$.

For all $a,b \in G'$

I If $b = x$ then by E6 $D_{M_2}(b) = D_M(b) - 1$ and $D_{M_1}(b) = D_{M|G'}(b) - 1$. If $b \neq x$ then by E6 $D_{M_2}(b) = D_M(b)$ and $D_{M_1}(b) = D_{M|G'}(b)$. In either case $D_{M_2}(b) = D_{M_1}(b) = D_{M_2|G'}(b)$.

II If $a = x = b$ or $a \neq x \neq b$ then $D_{M_2}(a|b) = D_M(a|b)$ and $D_{M_1}(a|b) = D_{M|G'}(a|b)$ If $a = x$ and $b \neq x$ then by E6 $D_{M_2}(a|b) = D_M(a|b) - 1$ and $D_{M_1}(a|b) = D_{M|G'} - 1$. In either case

$D_{M_2}(a|b) = D_{M_2|G'}(a|b)$. If $a \neq x$ and $b = x$ then by E6b if $D_{M_2}(a|b) \neq D_{M_1}(a|b)$ then since $D_{M_2}(a|b) \leq D_{M_1}(a|b)$, $D_{M_2}(a|b) = D_M(a|b)$ and $D_{M_1}(a|b) = D_{M|G'}(a|b) + 1$ hence $D_{M_1}(a|b) = D_M(b)$ and $D_{M|G'}(a|b) \neq D_{M|G'}(b)$ contradicting (1). Thus $D_{M_2}(a|b) = D_{M_2|G'}(a|b)$.

By I and II condition 1 is satisfied after a single vertex firing. Thus for any sequence σ of vertex firings $M|G'[\sigma\rangle M_1$ implies there exists M_2 such that $M[\rightarrow M_2$ and $M_2|G' = M_1$.

Q.E.D.

Theorem E7

In a marked graph with marking M, let L be any set of live vertices such that

(1) If $x \in L$ then for all y, $D_M(x|y) = 0$ implies $y \in L$.

There exists a firing sequence σ beginning with any firable vertex in L which fires every element of L exactly once and fires no other vertices. That is $x \in L \iff \|\sigma|x\| = 1$, $x \notin L \iff \|\sigma|x\| = 0$.

Proof:

We will first show that L contains a firable vertex. Pick any vertex $x \in L$. All vertices in the set B_x must also be in L since if $y \in B_x$, $D_M(x|y) = 0$. By the argument in E4, B_x contains a firable vertex.

Now let x be any firable vertex in L. Let $M[x\rangle M'$. Now kill x by placing a blank self loop on it: 

Now let $L' = \{y | y \in L \text{ and } y \neq x\}$. By E5 every vertex in L' is live. Since every output arc of x now contains at least one token, the set L' satisfies property (1). Thus the above procedure may be repeated for L'. By iteration every vertex in L can be fired exactly once without firing any other vertices. Q.E.D.

Corollary E7a

There exists a firing sequence beginning with any firable vertex which fires every live vertex of a marked graph exactly once.

Corollary E7b

There exists a firing sequence beginning with any firable vertex which fires every vertex of a live marked graph exactly once.

Theorem E8

A live graph remains live through firings.

Proof:

By E5 a vertex firing can only affect the liveness of the fired vertex. Assume the graph is live and $M[x\rangle M$. Assume x becomes dead as a result of the firing. Then relative to M $D_M(x) = 1$. Thus by E4 there must be a dead vertex d such that $\| \overrightarrow{d \ldots x} | M \| = 1$. However the graph is initially live and thus d cannot exist.

Q.E.D.

Theorem E9

In a live graph with marking M

$$M[-\rangle M' \Rightarrow M'[-\rangle M$$

Proof:

We will first show that for any vertex x,

$$M[x\rangle M_1 \Rightarrow M_1[-\rangle M$$

By E7, if $M[x\rangle M_1$, there exists a firing sequence σ beginning with x which fires every vertex in the graph exactly once. Since the initial and terminal vertex of every arc is fired once in this sequence, $M[\sigma\rangle$ is the same as M. Thus for any firable vertex x, $M[x\rangle M_1[-\rangle M$.

Now let σ_1 be any firing sequence such that $M[\sigma_1\rangle M'$.

Let the vertices of σ_1 be $x_1, x_2, x_3 \ldots x_n$ then $M[x_1\rangle M_1[x_2\rangle M_2[x_3\rangle \ldots [x_n\rangle M'$. By the preceding argument $M_n[x_{n+1}\rangle M_{n+1}$ implies $M_{n+1}[-\rangle M_n$. By repeated application $M'[-\rangle M$. Q.E.D.

Theorem E10

If the graph $\mathcal{G}$ with marking M is finite and connected and $M[-\rangle M$ then the graph is live.

Proof:

Let $M[\sigma\rangle M$. Let $x \in \sigma$ and $\|\sigma|x\| = N > 0$. We will show that every vertex is fired N times in σ , that is $\forall v \in \mathcal{G}$ $\|\sigma|v\| = N$.

Let v be any vertex. Since the graph is connected there exists a path P , not necessarily a directed path, from x to v . Since $M[\sigma\rangle = M$, $\|P|M[\sigma\rangle\| = \|P|M\|$. Thus by D5 $\|\sigma|x\| = \|\sigma|v\| = N$. Since every vertex is fired N times every vertex is live and $\mathcal{G}$ is live. Q.E.D.

Definition:

If M and T are markings, $M \geq T$ means: For all arcs α , $M(\alpha) \geq T(\alpha)$.

Theorem E11

Let G be a finite live marked graph with marking M. Let T be a marking of G. Let C be any circuit.

$$\left(\forall C \quad |C|M\| \geq \|C|T\|\right) \Longleftrightarrow \left(\exists M' \quad |M[->M' \text{ and } M' \geq T\right) .$$

Proof:

$\leftarrow$ Since $M' \geq T$ then $\forall a M'(a) \geq T(a)$. Thus $\forall C \quad |C|M'\| \geq \|C|T\|$. However, since $M[->M'$ and C has no endpoints, by D1, $\forall C \quad |C|M'\| = \|C|M\|$. Thus $\forall C \quad |C|M\| \geq \|C|T\|$.

$\rightarrow$ For any path P,

Define $d_i(P) \triangleq \begin{cases} d_i(P) = T(P) - M_i(P) & \text{if } T(P) > M_i(P) \\ d_i(P) = 0 & \text{if } T(P) \leq M_i(P) \end{cases}$

Let a be any arc,

Define $D_i = \sum_{a \in G} d_i(a)$

Clearly if $D_i = 0$ then $\forall a \quad T(a) \leq M_i(a)$ and $M_i \geq T$. We will show that there exists a sequence of markings $\mathfrak{M} = M[->M_1[->M_2 \ldots M_n$ such that $D_n = 0$. We will first prove that if $M_i \in \mathfrak{M}$ and $D_i = N > 0$ there exists M_{i+1} such that $D_{i+1} < D_i$. Let α be an arc for which $d_i(\alpha) > 0$ that is, $T(\alpha) > M_i(\alpha)$. Let $\dagger(\alpha, y)$ and $\dagger(\alpha, x)$.

Now let V be the set of all vertices v such that there exists a path $\overrightarrow{P\|vx}$ such that $d_i(P) > 0$.

(1) $y \notin V$. If $y \in V$ then there exists $\overrightarrow{P\|yx}$ such that $d_i(P) \neq 0$. However $\dot{\dagger}(\alpha,y)$, $\dot{\dagger}(\alpha,x)$ and $M_i(\alpha) < T(\alpha)$. Thus /there exists a circuit $\overrightarrow{C\|x,\alpha,y,x}$ such $d_i(C) \neq 0$, that is, $\|C|M\| < \|C|T\|$ contradicting hypothesis.

(2) If a is an arc such that $(\dot{\dagger}a) \in V$ and $(\dot{\dagger}a) \notin V$ then $M_i(a) > T(a)$. If $M_i(a) \leq T(a)$ then there exists a path $\overrightarrow{P\|a\ldots x}$ such that $d_i(P) \neq 0$ and thus $\dot{\dagger}a \in V$ contradicting assumption. Since $M_i(a) > T(a) \geq 0$, $M_i(a) > 0$. Thus if $d \notin V$ and $v \in V$ $\|\overrightarrow{\|d\ldots v|M_i\|} > 0$.

Now place a blank self-loop on all vertices not contained in V . By E6 all the vertices in V must still be live since by (2) above the token length of every path entering V from the set of dead vertices not in V is at least 1. By E7 there exists a sequence σ which fires every vertex in V exactly once. Let $M_i[\sigma\rangle M_{i+1}$ and a be any arc.

I If $(\dot{\dagger}a) \in V$ and $(\dot{\dagger}a) \in V$ or if $(\dot{\dagger}a) \notin V$ and $(\dot{\dagger}a) \notin V$ then $\|\sigma|\dot{\dagger}a\| = \|\sigma|\dot{\dagger}a\|$, thus $M_i(a) = M_{i+1}(a)$ and $d_i(a) = d_{i+1}(a)$.

II If $(\dot{\dagger}a) \in V$ and $(\dot{\dagger}a) \notin V$ then $\|\sigma|\dot{\dagger}a\| - \|\sigma|\dot{\dagger}a\| = 1$ and $M_{i+1}(a) = M_i(a) - 1$. By (2) $M_i(a) > T(a)$ and $M_{i+1}(a) = M_i(a) - 1 \geq T(a)$ thus $d_i(a) = d_{i+1}(a) = 0$.

III If $(\dot{\dagger}a) \notin V$ and $(\dot{\dagger}a) \in V$ then $\|\sigma|\dot{\dagger}a\| - \|\sigma|\dot{\dagger}a\| = 1$

and $M_{i+1}(a) = M_i(a) + 1$. Thus if $d_i(a) = 0$ then $d_{i+1}(a) = 0$ and if $d_i(a) = N \neq 0$ then $d_{i+1}(a) = N - 1$. Since there exists an arc, α , which satisfies requirement III and $d_i(\alpha) \neq 0$, then $d_{i+1}(\alpha) < d_i(\alpha)$. By I, II, and III for all arcs a , $d_{i+1}(a) \leq d_i(a)$. Thus,

$$\sum_{a \in G} d_{i+1}(a) < \sum_{a \in G} d_i(a) \quad \text{and} \quad D_{i+1} < D_i' .$$

Since $M[->M_i'$ for all M_i' in the marking μ , $\forall M_i \forall C \quad \|C|M_i\| \geq \|C|T\|$. Thus the hypothesis still holds and the above procedure may be repeated until D_i decreases to zero and $M_i \geq T$. Q.E.D.

Theorem Ella

The following theorem is a generalization of Theorem Ell to graphs which are not live. It's proof is very similar to that of Ell and differs in spots marked with asterisks in the margin.

** ← Example

Let G be a finite marked graph with marking M . Let T be a marking of G . Let C be any backwards track. $(\forall C \ \|C|M\| \geq \|C|T\|) \leftrightarrow (\exists M' | M[->M' \text{ and } M' \geq T)$

Proof:

← Since $M' \geq T$ then $\forall a M'(a) \geq T(a)$. Thus $\forall C \|C|M'\| \geq \|C|T\|$. However, since $M[\rightarrow M'$ and C has/beginning endpoint , by D1, $\forall C \|C|M'\| \leq \|C|M\|$. Thus $\forall C \|C|M\| \geq \|C|T\|$.

** no

→ For any path P ,

Define $d_i(P) \triangleq \begin{cases} d_i(P) = T(P) - M_i(P) & \text{if } T(P) > M_i(P) \\ d_i(P) = 0 & \text{if } T(P) \leq M_i(P) \end{cases}$

Let a be any arc,

Define $D_i = \sum_{a \in G} d_i(a)$

Clearly if $D_i = 0$ then $\forall a\ T(a) \leq M_i(a)$ and $M_i \geq T$. We will show that there exists a sequence of markings $\mathfrak{M} = M[\rightarrow M_1[\rightarrow M_2 \ldots M_n$ such that $D_n = 0$. We will first prove that if $M_i \in \mathfrak{M}$ and $D_i = N > 0$ there exists M_{i+1} such that $D_{i+1} < D_i$. Let α be an arc for which $d_i(\alpha) > 0$ that is, $T(\alpha) > M_i(\alpha)$. Let $\dot{\dagger}(\alpha,y)$ and $\dot{\dagger}(\alpha,x)$.

Now let V be the set of all vertices v such that there exists a path $\overrightarrow{P\|vx}$ such that $d_i(P) > 0$.

(1) y ∉ V . If $y \in V$ then there exists $\overrightarrow{P\|yx}$ such that $d_i(P) \neq 0$ and since for the arc α from x to y

** $M_i(\alpha) < T(\alpha)$. There exists a track (also a circuit) $\overrightarrow{C\|x,\alpha,y,x}$ such that $d_i(c) \neq 0$, that is, $\|C|M\| < \|C|T\|$ contradicting hypothesis.

(2) If a is an arc such that $(\dot{\dagger}a) \in V$ and $(\dot{\dagger}a) \notin V$ then $M_i(a) > T(a)$. If $M_i(a) \leq T(a)$ then there

exists a path $\overrightarrow{P\|a\ldots x}$ such that $d_i(P) \neq 0$ and thus $\dot{\uparrow}a \in V$ contradicting assumption. Since $M_i(a) > T(a) \geq 0$, $M_i(a) > 0$. Thus if $d \notin V$ and $v \in V$ $\|\overrightarrow{d\ldots v}|M_i\| > 0$.

Now place a blank self-loop on all vertices not
** contained in V. Suppose there exists $d \in V$ such that d is dead. Then by Theorem D4 there exists a backward blank track Cd ending at d. However $Cd \cup \overrightarrow{P\|v\ldots x}$ is also a track and $|Cd \cup \overrightarrow{P\|v\ldots x}|M\| = \|P|M\| < \|P|T\|$ contradicting the hypothesis. Thus all vertices in V are live. By E7 there exists a sequence σ which fires every vertex in V exactly once. Let $M_i[\sigma\rangle M_{i+1}$ and a be any arc.

I If $(\dot{\uparrow}a) \in V$ and $(\uparrow a) \in V$ or if $(\dot{\uparrow}a) \notin V$ and $(\uparrow a) \notin V$ then $|\sigma|\dot{\uparrow}a\| = \|\sigma|\uparrow a\|$, thus $M_i(a) = M_{i+1}(a)$ and $d_i(a) = d_{i+1}(a)$.

II If $(\dot{\uparrow}a) \in V$ and $(\uparrow a) \notin V$ then $|\sigma|\dot{\uparrow}a\| - \|\sigma|\uparrow a\| = 1$ and $M_{i+1}(a) = M_i(a) - 1$. By (2) $M_i(a) > 0$ and $M_{i+1}(a) = M_i(a) - 1 \geq 0$ thus $d_i(a) = d_{i+1}(a) = 0$.

III If $(\dot{\uparrow}a) \notin V$ and $(\uparrow a) \in V$ then $\|\sigma|\uparrow a\| - \|\sigma|\dot{\uparrow}a\| = 1$ and $M_{i+1}(a) = M_i(a) + 1$. Thus if $d_i(a) = 0$ then $d_{i+1}(a) = 0$ and if $d_i(a) = N \neq 0$ then $d_{i+1}(a) = N - 1$.

Since there exists an arc, α, which satisfies requirement III and $d_i(\alpha) \neq 0$, then $d_{i+1}(\alpha) < d_i(\alpha)$. By I, II, and III for all arcs a, $d_{i+1}(a) \leq d_i(a)$. Thus,

$$\sum_{a \in G} d_{i+1}(a) < \sum_{a \in G} d_i(a) \quad \text{and} \quad D_{i+1} < D_i' .$$

** Now we will prove that our hypothesis still holds.

Assume there exists a track C such that $\|C|M_{i+1}\| < \|C|T\|$. By assumption $\|C|M_i\| \geq \|C|T\|$. However, σ such that $M_i[\sigma>M_{i+1}$ the property that $\forall$ vertices a, $\|\sigma|a\| = \|V|a\|$ which is always 0 or 1. Thus since $\|C|M_i\| > \|C|M_{i+1}\|$ and the endpoint e of c is fired at most once in σ, $\|C|M_i\| = \|C|M_{i+1}\| + 1$. However $\|C|M_i\| \geq \|C|T\| > \|C|M_{i+1}\|$, thus $\|C|M_i\| = \|C|T\|$. But since $\|\sigma|e\| = 1$, $e \in V$. Thus $C \cup \overrightarrow{P\|e...x}$ is a track terminating at x and $\|C \cup \overrightarrow{P\|e...x}|M_i\| < \|C \cup \overrightarrow{P\|e...x}|T\|$ contradicting our initial assumption. Thus the hypothesis still holds and the above procedure may be repeated until D_i decreases to zero and $M_i \geq T$.

Q.E.D.

Corollary E12:

In a live finite strongly connected graph

$\left[\forall C \quad \|C|M_1\| = \|C|M_2\|\right] \iff M_1[->M_2$ (where C is any circuit).

Proof:

$\leftarrow$ Follows from D1.

$\rightarrow$ By E11 there exists $M_3 \mid M_1[->M_3$ and $M_3 \geq M_2$. Also by D1:

(1) $\forall C \ \|C|M_1\| = \|C|M_3\| = \|C|M_2\|$. Since the graph is strongly connected, every arc is contained in a circuit. If a is an arc, for any path P from $(\dot{\uparrow}a)$ to $(\dot{\uparrow}a)$. $\|P|M_3\| \geq \|P|M_1\|$. Thus if $M_3(a) > M_2(a)$ then there exists a circuit C composed of a and P such that $\|C|M_3\| > \|C|M_2\|$, contradicting (1). Thus $M_3(a) \leq M_2(a)$, $M_3(a) \geq M_2(a)$ and $\forall a \ M_3(a) = M_2(a)$. Therefore $M_3 = M_2$.

Q.E.D.

Definition

A cycle is any path $\overline{P\|x\ldots x}$, not necessarily directed.

Theorem E13:

Let g be the live finite graph $\langle g,M\rangle$. Let C be any cycle. $\left[\forall C \ \|C|M\| = \|C|M_2\|\right] \longleftrightarrow \left(M[\rightarrow M_2\right)$

$\leftarrow$ Follows from D1.

$\rightarrow$ By E11 there exists $M_1|M[\rightarrow M_1$ and $M_1 \geq M_2$. By D1.

(1) $\forall C \ \|C|M\| = \|C|M_1\| = \|C|M_2\|$.

Let $x,y \in g$. Let P and P' be any two paths from x to y. Since $P \oplus P'$ is a cycle,

$$M_1(P) - M_1(P') = M_2(P) - M_2(P')$$

$$M_1(P) - M_2(P) = M_1(P') - M_2(P')$$

Thus $\forall P$, $M_1(P) - M_2(P)$ is a constant. Now add an arc a to the graph from y to x. Let $M_1(a) > 0$ and $M_2(a) > 0$ and $M_2(a) - M_1(a) = M_1(P) - M_2(P)$ then $\forall P \ M_2(a) + M_2(P) = M_1(a) + M_1(P)$.

Thus all new cycles C satisfy the relation $M_1(C) = M_2(C)$. Place such an arc between every pair of vertices which are connected by a path. The graph now consists of strongly connected components.

By E12 $M_1[\rightarrow M_2$ and by D6 $M_1|g[\rightarrow M_2|g$.

Q.E.D.

F. Safety

We now introduce an important property of marked graphs which relates to the possible token content of arcs.

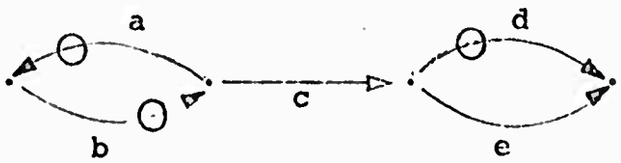
Definition F1

Define $U_M(\alpha) \triangleq \max_{M' \in \vec{M}} \|\alpha|M'\|$

$L_M(\alpha) \triangleq \min_{M' \in \vec{M}} \|\alpha|M'\|$

Example:

$\langle g, M \rangle$



$U_M(a) = 2$

$U_M(c)$ is undefined

$L_M(b) = 0$

$L_M(d) = 1$

Theorem F2

$U_M(\alpha) = M(\alpha) + D_M(\dot{\uparrow}\alpha|\dot{\uparrow}\alpha)$

$L_M(\alpha) = M(\alpha) - D_M(\dot{\uparrow}\alpha|\dot{\uparrow}\alpha)$

Proof:

$D_M(\dot{\uparrow}\alpha|\dot{\uparrow}\alpha) = N$ if and only if $\max_{M[\sigma\rangle M'} (\|\sigma|\dot{\uparrow}\| - \|\sigma|\dot{\uparrow}\|) = N$

Clearly $U_M(\alpha) = M'(\alpha) = M(\alpha) + N$. By similar argument $L_M(\alpha) = M(\alpha) - D_M(\dot{\uparrow}\alpha|\dot{\uparrow}\alpha)$.

Theorem F3

$$U_M(\alpha) = \min_T \left\| \overleftarrow{T \| \alpha \ldots} \right| M \|$$

Proof: By F2 $U_M(\alpha) = M(\alpha) + D_M(\dot{\uparrow}\alpha | \dot{\uparrow}\alpha)$

Assume $D_M(\dot{\uparrow}\alpha | \dot{\uparrow}\alpha) = D_M(\dot{\uparrow}\alpha)$.

Then $\min_T \left\| \overleftarrow{T \| \dot{\uparrow}\alpha \ldots} \right| M \| = D_M(\dot{\uparrow}\alpha) = D_M(\dot{\uparrow}\alpha | \dot{\uparrow}\alpha)$

Clearly $\min_T \left\| \overleftarrow{T \| \alpha \ldots} \right| M \| = D_M(\dot{\uparrow}\alpha | \dot{\uparrow}\alpha) + M(\alpha) = U_M(\alpha)$.

Now assume $D_M(\dot{\uparrow}\alpha | \dot{\uparrow}\alpha) \neq D_M(\dot{\uparrow}\alpha)$. Then

$\min_P \left\| \overleftarrow{P \| \dot{\uparrow}\alpha \ldots \dot{\uparrow}\alpha} \right| M \| = D_M(\dot{\uparrow}\alpha | \dot{\uparrow}\alpha)$. The circuit composed of P and the arc α is a track thus

$\min_T \left\| \overleftarrow{T \| \alpha \ldots} \right| M \| = D_M(\dot{\uparrow}\alpha | \dot{\uparrow}\alpha) + M(\alpha) = U_M(\alpha)$.

Q.E.D.

Definition:

The arc α is safe $\triangleq$ $U_M(\alpha) \leq 1$. A graph is safe if all its arcs are safe.

Definition:

A basic circuit is a circuit C such that $\|C | M\| = 1$

Theorem F4

If $\mathcal{N}$ is live and $\alpha \in \mathcal{N}$

α is safe if and only if it is contained in a basic circuit.

Proof:

$\rightarrow$ By F3 α must be contained in a track with one token

or less. If this track is not a circuit then it contains a dead vertex contradicting assumption.

If it is a circuit C then since the graph is live $\|C|M\| \neq 0$. Thus $\|C|M\| = 1$.

If $\|C|M\| = 1$ and $\alpha \in C$ then by F3 $U_M(\alpha) \leq 1$. Since $\uparrow\alpha$ is live $U_M(\alpha) = 1$ and the arc is safe.

Q.E.D.

Theorem F5

Given a live graph $M_0 = \langle g, M \rangle$ and an unsafe arc α contained in a circuit. There exists M' such that $\langle g, M' \rangle$ is live, α is safe in M' and every arc which is safe in M is safe in M'.

Proof: (Genrich)

Since α is contained in a circuit, $U_M(\alpha)$ is defined. Let $M[->M_1$ and $M_1(\alpha) = U_M(\alpha)$. Since the graph is live and for all circuits C $\|C|M\| = \|C|M_1\|$, and $M_1(\alpha) = U_M(\alpha)$ $U_M(\alpha) = 1$ implies $U_{M_1}(\alpha) = 1$.

Define M' : For all arcs $a \neq \alpha$, $M'(a) = M(a)$ and $M'(\alpha) = 1$

Since $M_1(\alpha) = U_M(\alpha)$, $U_{M'}(\alpha) = 1$ and α is safe.

For all circuits $\|C|M_1\| \geq \|C|M'\| > 0$ thus all arcs safe in M_1 are safe in M' and M' is live.

Corollary F6

Any finite graph g composed of strongly connected

components may be given a live safe marking M .

Proof:

For all arcs, let $M_0(a) \geq 0$. Clearly $\langle g, M_0 \rangle$ is live. Since every arc is contained in a circuit, the above procedure may be performed for each arc resulting in a live safe marking.

Q.E.D.

Distances

Define $D(x|y) \triangleq \max_{M[\rightarrow M'} D_{M'}(x|y)$

Theorem F7

Let x and y be two vertices in a live graph $\mathcal{G}$. Let Cxy be any circuit containing x and y . Then $D(x|y) = \min_{C} \|Cxy|M\|$

Proof:

First we will show that $D(x|y) \leq \min_{C} \|Cxy|M\|$.

Assume $M[\rightarrow M'$. Since g is live $D_{M'}(x|y) = \min_{P} \|\overleftarrow{P}\overleftarrow{x...y}|M\|$. Clearly $\min_{P} \|\overleftarrow{P}\overleftarrow{x...y}|M\| \leq \min_{C} \|Cxy|M\|$. Thus $\max_{M[\rightarrow M'} D_{M'}(x|y) \leq \min_{C} \|Cxy|M\|$ and $D(x|y) \leq \min_{C} \|Cxy|M\|$.

Now we will prove that there exists M' such that

$M[\rightarrow]M'$ and $D_{M'}(x|y) = \min_C \|Cxy|M\|$

Let $D_M(x|y) = N$ and $D_M(y|x) = K$

Now let $M[\sigma\rangle M_1$ and $\|\sigma|y\text{-}x\| = K$. Then $D_{M_1}(y|x) = 0$ and $D_{M_1}(x|y) = N + K$. Furthermore $\min_P \|\overleftarrow{P}|y\ldots x|M\| = 0$ and $\min_P \|\overleftarrow{P}|x\ldots y|M\| = N + K$ and thus

$\min_C \|Cxy|M\| = N + K = D_{M_1}(x|y)$.

Q.E.D.

Corollary F8

If $\langle \vartheta, M\rangle$ is live, $D(x|y)$ has the following properties:

1. $\forall x,\ D(x|x) = 0$
2. $\forall x,y\ D(x|y) = D(y|x)$
3. $\forall x,y,z\ D(x|z) \leq D(x|y) + D(y|z)$

Proof:

(1) follows from definition

(2) By F7 $D(x|y) = \min_C \|C_{xy}|M\|$

$D(y|x) = \min_C \|C_{xy}|M\|$ thus

$D(x|y) = D(y|x)$

(3) By F7,

$D(x|z) = \min_C \|C_{xz}|M\|$

$D(x|y) = \min_C \|C_{xy}|M\|$

$D(y|z) = \min_C \|C_{yz}|M\|$

Every circuit pair C_{xy} and C_{yz} defines a circuit C_{xyz} consisting of $C_{xy} \oplus C_{yz}$ such that

$\|C_{xyz}|M\| = \|C_{xy}|M\| + \|C_{yz}|M\|$.

Thus $\min_C \|C_{xz}|M\| \leq \min_C \|C_{xy}|M\| + \min_C \|C_{yz}|M\|$

Thus $D(x|z) \leq D(x|y) + D(y|z)$.

Q.E.D.

Theorem F10

In a live strongly connected graph $\gamma = \langle g,M \rangle$

Let V be a set of vertices such that $\forall x,y, \varepsilon V$ $D(x|y) = 1$. Then there exists a basic circuit containing all vertices in V .

Proof:

Let σ be any sequence which fires every vertex in the graph exactly once. Let $\sigma = s_1, s_2, s_3, \ldots s_n$. Let $M[\sigma\rangle M_1$. Now if $s_i, s_j' \in V$ and $i < j$, $D_{M'}(s_i|s_j) = 0$ and thus there is a blank path from s_j to s_i. If $i > j$, $D_{M'}(s_i|s_j) = 1$ and there is a path with one token from s_j to s_i. Now let $v_1, v_2, v_3 \ldots v_n$ be the vertices of V in the order in which they appear in σ. By the above argument for all i there is a blank path from v_i to v_{i+1} and a path with one token from v_n to v_1. Thus there exists a basic circuit $C = \overrightarrow{v_1, v_2, v_3 \ldots v_n v_1}$ containing V.

Q.E.D.

Theorem F11

Let C be a circuit in a live and safe graph $\mathcal{G}$

$$0 < \|C|M\| < \|C|A\|$$

Proof:

Since $\mathcal{G}$ is live $\|C|M\| > 0$. Now assume $\|C|M\| \geq \|C|A\|$

Let $C = \overrightarrow{\|a_1,a_2,a_3 \ldots a_n a_1}$. Since the graph is live and safe, each arc is covered by a basic circuit. Thus for any arc $a_i \in C$ there exists a path $\overrightarrow{P_i \| \dot{t}a \ldots \dot{t}a}$ such that $\|P_i|M\| = 0$ if $M(a_i) = 1$ and $\|P|M\| = 1$ if $M(a_i) = 0$. Now let C' be the circuit consisting of $P_n, P_{n-1}, P_{n-2} \ldots P_1$. By the above argument $\|C'|M\| + \|C|M\| = \|C|A\|$ and $\|C'|M\| = \|C|A\| - \|C|M\|$. However by assumption $\|C|A\| - \|C|M\| \leq 0$ contradicting the hypothesis that the graph is live. Thus $\forall C \quad \|C|A\| > \|C|M\| > 0$.

Q.E.D.

Live and Safe Marked Graphs

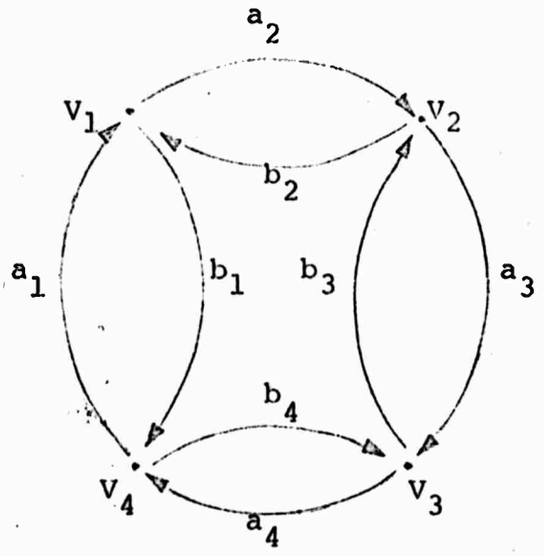
Definition:

A ls-graph is a live and safe marked graph.

Example 1: Billiard Balls

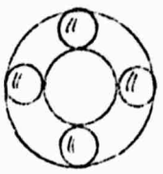
Consider the ls-graph $g_a \oplus g_b$ where g_a is the single circuit $\overrightarrow{C_a \| a_1, v_1, a_2, v_2 \ldots a_n v_n}$ and g_b is the single circuit $\overrightarrow{C_b \| b_n, v_{n-1}, b_{n-1}, v_{n-2} \ldots b_2, v_1, b_1, v_n}$.

Example for N = 4:



Such a graph may be interpreted as a representation of n billiard balls in ideal collisions on a functionless circular track. Each circuit $C_i = \overrightarrow{\|a_i, b_i}$ represents a billiard ball $\textcircled{\scriptsize{ii}}_i$ with two possible velocities, clockwise and counter-clockwise. A token on a_i means $\textcircled{\scriptsize{ii}}_i$ is moving clockwise; a token on b_i means $\textcircled{\scriptsize{ii}}_i$ is moving counterclockwise. Vertex firings represent collisions between balls.

N = 4

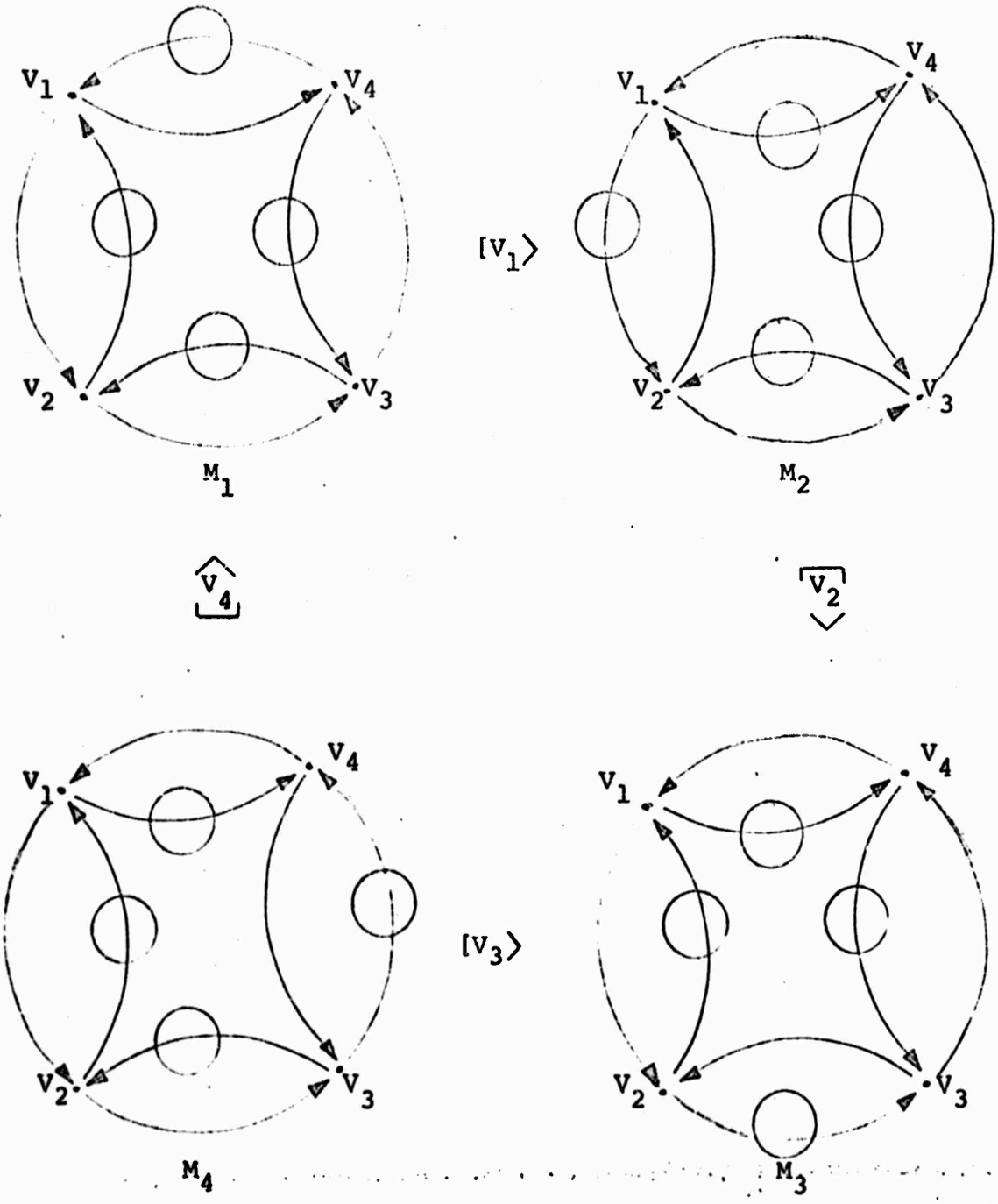


Different marking classes of the graph represent different modes of bouncing by which the balls may interact. Now consider any circuit $\overrightarrow{C_i = \|a_i, b_i}$. By F11 $0 < \|C_i|M\| < \|C_i|A\|$ $\|C_i|A\| = 2$ thus $\|C_i|M\| = 1$. (This makes sense if C_i is to represent $\textcircled{\scriptsize ⫴}_i$ which can only move in one direction at a time.)

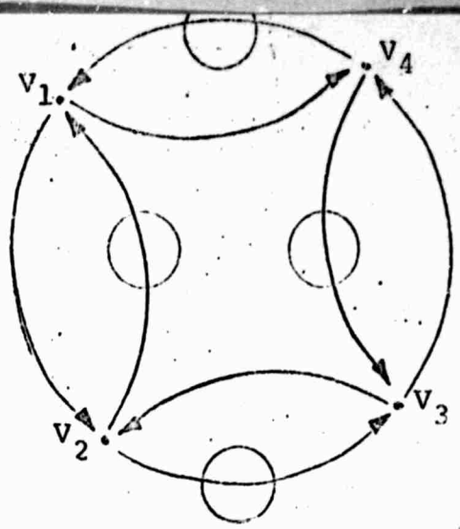
Since the C_i circuits cover the graph the total number of tokens in any marking will be n. By E12, to completely specify the marking class of a given marking, we need only indicate how the n tokens are distributed on the outer and inner simple circuits C_a and C_b. Clearly $\|C_a + C_b|M\| = n$. $0 < C_a < n$ thus the number of inequivalent marking classes is $n - 1$. If $\mathfrak{M}_i$ is the marking class which puts i tokens on C_a, the number of markings $M \in \mathfrak{M}_i$ is the number of different ways of distributing the i tokens on C_a. Since $\|C_a|A\| = n$ this number is $\binom{n}{i}$.

Example: $\mathfrak{M}_1 : \binom{4}{1} = 4$ markings

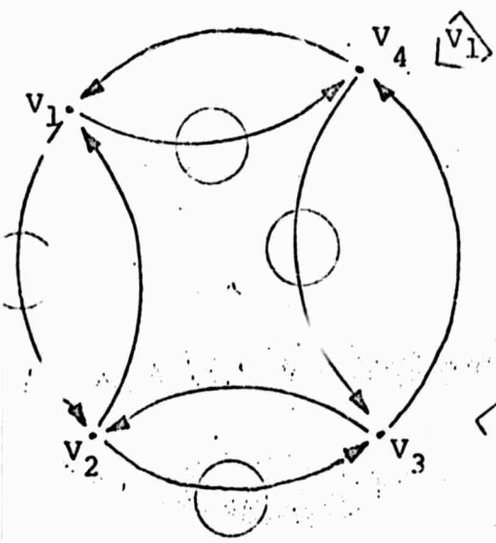
$\mathfrak{M}_1$ represents the bouncing mode in which three balls are moving clockwise and one ball is moving counter-clockwise.



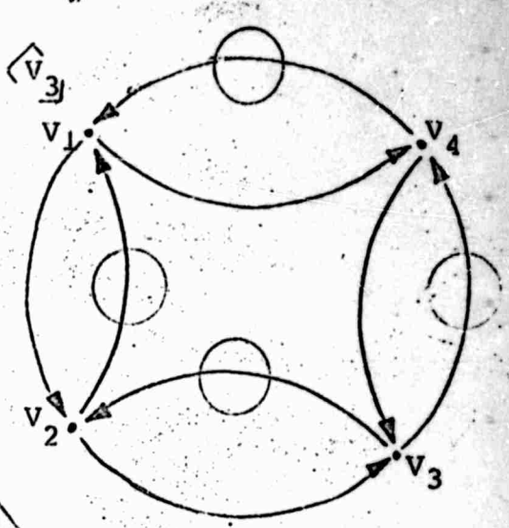
$\Pi_2 : \binom{4}{2} = 6$ markings



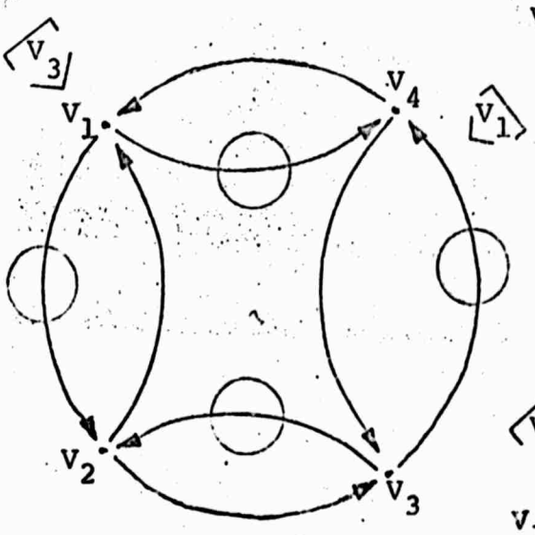
M_1



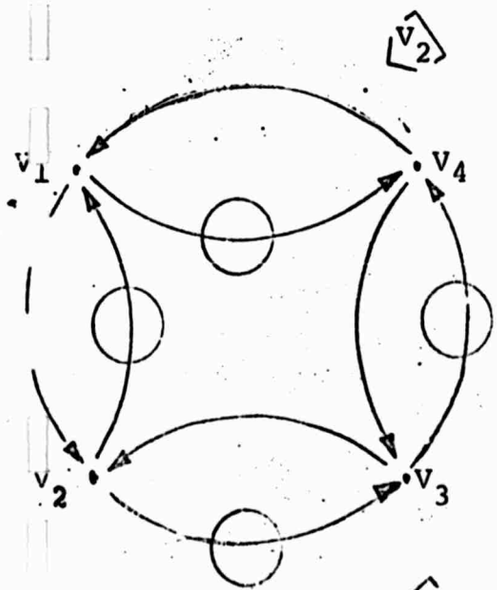
M_3



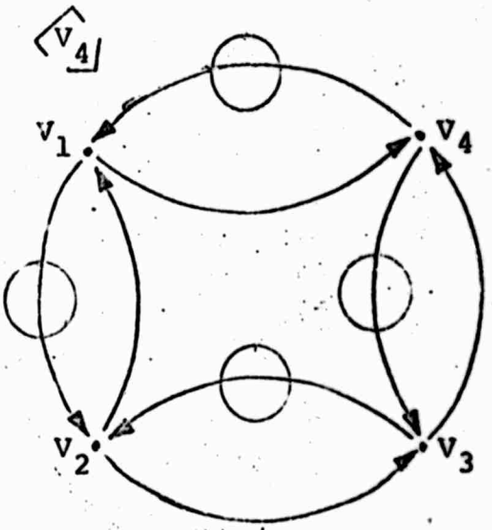
M_2



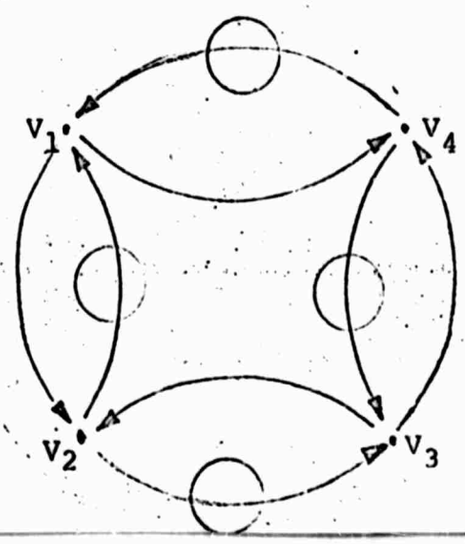
M_4



M_5



M_6



M_1

$\mathcal{M}_2$ represents the bouncing mode in which two balls are moving clockwise and two balls are moving counter-clockwise. Note that the total number of possible ways of placing one token on each two-arc circuit C_i is 2^n. The number of these distributions which are live and safe is

$\sum_{i=1}^{i=n-1} \binom{n}{i} = 2^n - 2$ leaving two distributions

$C_a = 0$, $C_b = n$; and $C_a = n$, $C_b = 0$; which are safe but not live. These two distributions represent the bouncing modes where all balls are moving clockwise or all balls are moving counter-clockwise, hence no collisions occur and hence no vertex is live.

Example 2

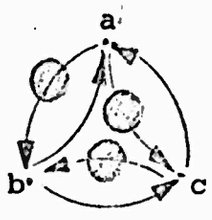
Consider a graph with N vertices with an arc from every vertex to every vertex. This graph is called the complete graph on N vertices. Any safe marking of this graph will cover every arc with a basic circuit. Thus for any live and safe marking, for all vertices x and y, $D(x|y) = 1$. By F10 there must exist a basic circuit containing all the vertices of the graph. Now let M be a live and

safe marking of the graph. Since $\forall x,y D(x|y) = 1$, no two vertices may be firable concurrently. Thus there exists exactly one x such that $M[x\rangle$ exists. Now let M_1 and M_2 be live and safe markings of the graph in which x is firable. If $M_1[\rightarrow M_2$ then M_1 and M_2 contain the same basic circuit containing all vertices. In M_1 and M_2 there is a directed blank path for all arcs α either from $\dot{\alpha}$ to $\dot{\alpha}$ or from $\dot{\alpha}$ to $\bar{\alpha}$. Thus there exists a cycle C in M_1 and M_2 such that $M_1(C) = M_1(\alpha)$ and $M_2(C) = M_2(\alpha)$. Since $M_1(C) = M_2(C)$ $M_1(\alpha) = M_2(\alpha)$ thus $M_1 = M_2$. Hence there is exactly one marking per $\vec{M}$ which makes any given vertex firable. Thus the number of markings in any $\vec{M}$ is N. Each $\vec{M}$ is a full cyclic ordering of the N vertices where the ordering is expressed in the order of the vertices as they appear on the basic circuit, that is, the order in which they fire. Furthermore, any full cyclic ordering is expressed in some $\vec{M}$. Thus the number of marking classes $\vec{M}$ is the number of inequivalent cyclic full orderings of N elements which is $\frac{N!}{N} = (N-1)!$

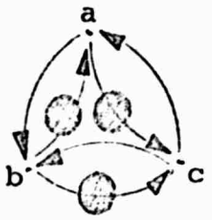
Example:

N = 3 (N-1)! = 2

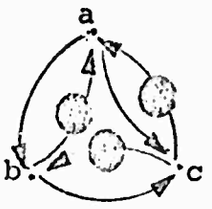
m



M_1

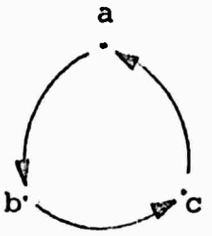


M_2

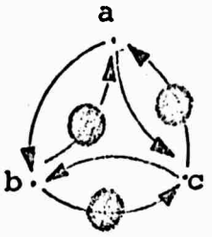


M_3

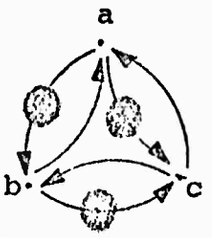
Full cyclic ordering:



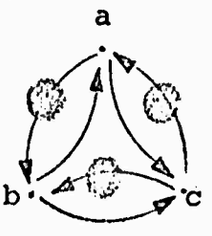
m'



M'_1

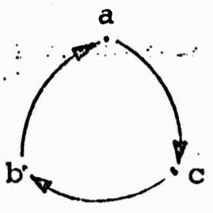


M'_2



M'_3

Full cyclic ordering:



G. Boundary and Coboundary

In this section, we introduce notation which allows us to generate theorem D1.

This theorem states that the number of tokens on a path increases by only 1 when the initial vertex of the path fires and decreases by 1 only when the terminal vertex fires.

A generalization of this fact to structures more complicated then paths enables us to determine what vertex firings affect the token content of such structures.

Definition

The boundary of the arc α is $\dot{\uparrow}(\alpha) - \dot{\uparrow}(\alpha)$.

We write $\delta(\alpha) = \dot{\uparrow}(\alpha) - \dot{\uparrow}(\alpha)$

If A and B are linear expressions in arcs,

$$\delta(A+B) \triangleq \delta(A) + \delta(B)$$

Examples:

The boundary of a path $\overrightarrow{P\,|x \ldots y}$ expressed as the sum of its arcs is $x - y$. The boundary of a cycle is 0 .

Definition:

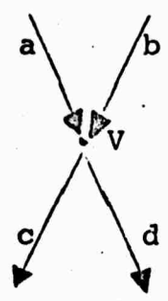
The coboundary of the vertex v is

$$\sum_{\dot{\uparrow}(a,v)} a \;-\; \sum_{\dot{\uparrow}(b,v)} b \;=\; \partial v$$

If U and V are linear expressions in vertices

$$\partial(U+V) \triangleq \partial(U) + \partial(V)$$

Example:



$$\partial V = c + d - a - b$$

Theorem G1:

Let $M[\sigma\rangle M'$. $M' = M + \partial\sigma$.

Proof:

If σ is a single vertex firing, x , $M[x\rangle$ differs from M only in that $M[x\rangle$ places one more token on the outputs of x and one less on the inputs, that is, $M[x\rangle = M + \partial x$. Then if $M\ \Sigma\ x_1, x_2, x_3 \ldots\rangle M^1$, $M^1 = M + \partial x_1 + \partial x_2 \ldots \partial x_n = M + \partial\sigma$.

Theorem G2:

Let E be an expression in arcs and σ be a firing sequence such that $M[\sigma\rangle M'$

$$\|E|\partial\sigma\| = \|\sigma|\delta E\|$$

Proof:

$$\|E|\partial\sigma\| = \|E|M^1 - M\|$$

If $E = a_1x_1 + a_2x_2 \ldots a_nx_n$

$$\|E|\partial\sigma\| = a_1\|x_1|\partial\sigma\| + a_2\|x_2|\partial\sigma\| \ldots a_n\|x_n|\partial\sigma\|$$

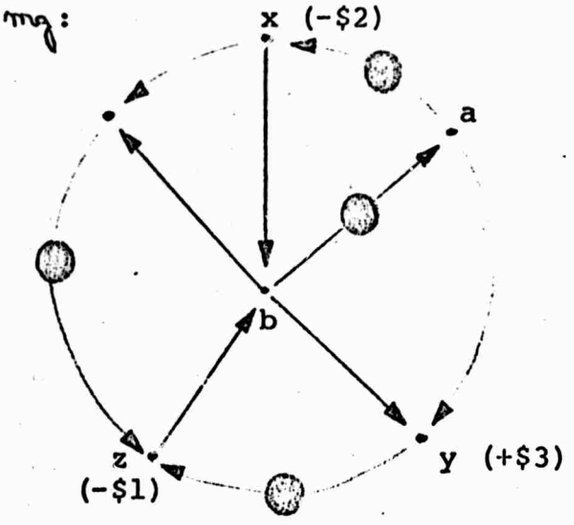
However $\quad a\|x|\partial\sigma\| = a(\|\sigma|\dot{\uparrow}x\| - \|\sigma|\dot{\downarrow}x\|) = a\|\sigma|\delta x\|$

Since $\quad E = \sum_i a_i x_i \qquad \|E|\partial\sigma\| = \sum_i a_i \|\sigma|\delta x_i\|$

$$\|\sigma|\textstyle\sum_i a_i x_i\| = \|\sigma|\delta E\|$$

Maximal and Minimal Firing Sequences

Theorem E6b states that $\min_{P} \|P\overrightarrow{\|y...x}|M\| = \max_{M[\sigma\rangle} \|\sigma|x-y\|$
This theorem allows us to determine the maximum size, relative to a given measure $\|\sigma|x-y\|$, of a firing sequence beginning at M. Suppose we wish to find the size of a maximum sequence relative to some more complicated measure, for example $\max_{M[\sigma\rangle} \|\sigma|2x-3y+z\|$. Such an expression may represent a meaningful quantity:



This marked graph represents the operating constraints on a business. Vertex firings are financial events. An arc α from vertex p to vertex q means that the event q requires some commodity which is produced by the event p. Tokens on α represent the present supply of that commodity.

Assume x represents an event which costs the business $2. Event z costs $1 and event y brings a profit of $3. If a firing sequence σ represents a sequence of events in the business $\$\|\sigma|2x-3y+z\|$ is the net amount of cash

spent by the business in this sequence. Thus $\max_{M[\sigma\rangle} \|\sigma|2x-3y+z\|$ is the minimum initial capital which insures that the business, under the operating constraints expressed by $\mathcal{N}$, does not go into the red. In the example, the firing sequence $\sigma = (z,a,x,x,b,a,x)$ yields a \$7 debt which is believed maximal.

In this section we will show how the sizes of such maximal firing sequences may be found by inspection of the graph at its initial marking M.

Theorem G3:

In a strongly connected live graph g

Let U be a linear expression in vertices such that there exists an expression E in arcs with positive coefficients such that $-\delta E = U$

$$\max_{M[\sigma\rangle} \|\sigma|U\| = \min_{-\delta E=U} \|E|M\|$$

Proof:

First we will show that

$$\max_{M[\sigma\rangle} \|\sigma|U\| \leq \min_{-\delta E=U} \|E|M\|$$

Let σ be any sequence such that $M[\sigma\rangle M'$. By G1 and G2 since $-U = \delta E$, $\|E|M\| - \|\sigma|U\| = \|E|M'\|$ thus $\|\sigma|U\| = \|E|M\| - \|E|M'\|$. However since E has positive coefficients $\|E|M'\| \geq 0$ thus for all $M[\sigma\rangle$ and for all E such that $-\delta E = U$, $\|\sigma|U\| \leq \|E|M\|$. Thus in particular $\max_{M[\sigma\rangle} \|\sigma|U\| \leq \min_{-\delta E=U} \|E|M\|$.

Now we will show that there exists $M[\sigma\rangle$ and there exists E such that $-\delta E = U$ and $\|\sigma|U\| = \|E|M\|$.

For a forwards directed path p, define

$b(p) \triangleq$ the beginning point of p

$e(p) \triangleq$ the ending point of p

Clearly $\delta P = b(p) - e(p)$

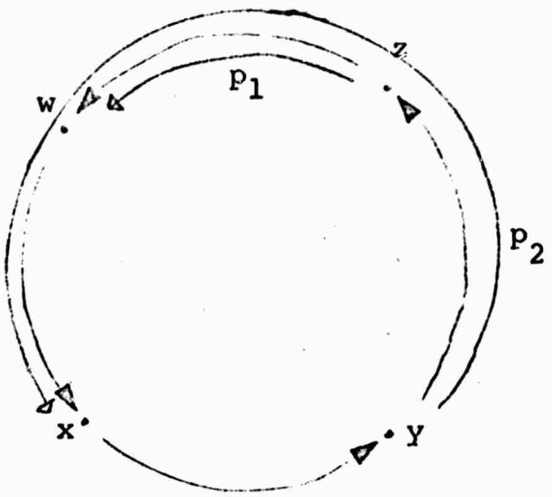
Let P be a set of forward directed paths $\{p_1 \ldots p_n\}$

A) such that $\sum_i \delta(p_i) = -U$ and $\sum_i \|p_i|M\|$ is minimal.

Since g is strongly connected such a set P may be constructed by choosing p_1 such that $\|U|b(p_1)\|$ is negative and $\|U|e(p_1)\|$ is positive. Now define $U_o = U$ $U_i \triangleq U_{i-1} + \delta p_i = U_{i-1} + b(p_i) - e(p_i)$ and choose p_i such that $\|U_i|b(p_i)\|$ is negative and $\|U_{i-1}|e(p_i)\|$ is positive. Since there exists E

such that $-\delta E = U$, and since E is made up of expressions of the form $\dot{\phi}(\alpha) - \dot{\phi}(\alpha)$, the sum of the positive coefficients in U is equal to the sum of the negative coefficients in U. Thus in the path construction procedure described we will not run out of initial endpoints before we have used all terminal endpoints. Thus the set P exists.

Example:

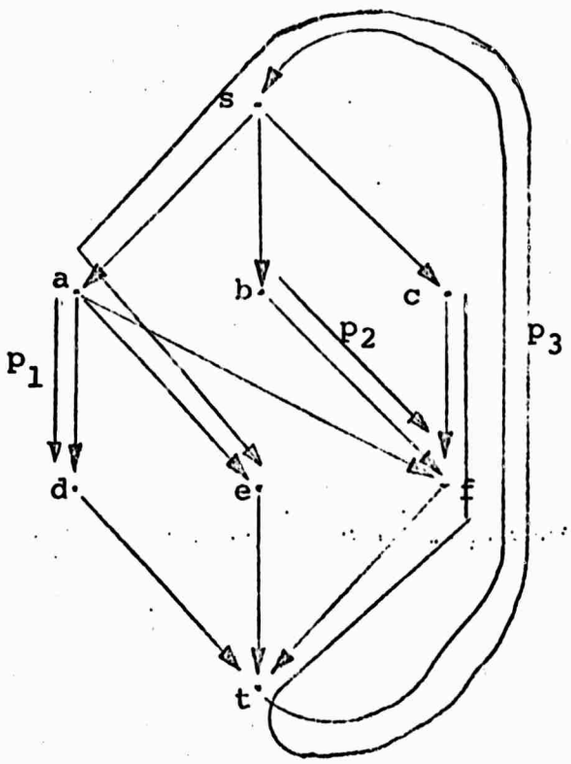


$U = w + x - (y+z)$

$P = \{p_1, p_2\}$

$\overrightarrow{p_1 | zw}$

$\overrightarrow{p_2 | yx}$



$U = d + e + f - (a+b+c)$

$P = \{p_1, p_2, p_3\}$

$\overrightarrow{p_1 | ad}$

$\overrightarrow{p_2 | bf}$

$\overrightarrow{p_3 | cftsae}$

Now this set P has the property that $-\delta P = U$ ($-\sum_i \delta(p_i) = U$) . We wish to show that there exists σ such that $M[\sigma\rangle M'$ and $\|\sigma|U\| = \|P|M\|$.

Since $U = -\delta P$, this is equivalent to $\|P|M'\| = 0$.

Thus we need only show that there exists M' such that $M[\rightarrow M'$ and $\|P|M'\| = 0$. To show this we will construct an expression E such that $\delta E = U = -\delta P$ and show that there exists M' such that $M[\rightarrow M'$ and $\|E|M'\| = \|P|M\| + \|E|M\|$ .However since $\delta E = -\delta P$, $\delta(E+P) = 0$, thus $\|P|M\| + \|E|M\| = \|P|M'\| + \|E|M\|$ and hence $\|P|M'\| = 0$.

First add to g a set of arcs $\{\alpha_1 \ldots \alpha_n\}$ such that $\dot{\uparrow}(\alpha_i) = b(p_i)$, $\dot{\downarrow}(\alpha_i) = e(p_i)$, and $M(\alpha_i) = 1$. Since $M(\alpha_i) > 0$ the graph is still live. By E6c every firing sequence in the new graph is also a firing sequence in g . Clearly $\sum_\alpha \delta(\alpha) = -\delta P$.

Now define a test marking T such that

For all α_i $T(\alpha_i) = M(\alpha_i) + \|p_i|M\|$

For all arcs $a \notin \{\alpha_1, \alpha_2 \ldots \alpha_n\}$ $T(a) = 0$

We wish to show that there exists M' such that $M[\rightarrow M'$ and $M' \geq T$

Assume this is not the case.

Then by E11 there exists a circuit C such that $\|C|M\| < \|C|T\|$. since $a \notin \{\alpha_i\}_i$ implies $T(a) = 0$, C must include some α_i arcs.

Let $C = \overrightarrow{\|\alpha_1 \ldots \alpha_2 \ldots \alpha_j}$ (new subscripts)

Let p_i' be the path in C from α_i to α_{i+1} (p_j' goes from α_j to α_1) .

By assumption

$\|C|M\| < \|C|T\|$ thus

$$\sum_{i=1}^{j} M(\alpha_i) + \sum_{i=1}^{j} M(p_i') < \sum_{i=1}^{j} T(\alpha_i) + \sum_{i=1}^{j} T(p_i')$$

But T only places tokens on α_i arcs, thus for all p' T(p') = 0 and/ we have $\sum M(\alpha_i) + \sum M(p_i') < \sum T(\alpha_i)$.

By definition

$T(\alpha_i) = M(\alpha_i) + M(p_i)$ thus

$\sum M(\alpha_i) + \sum M(p_i') < \sum M(\alpha_i) + \sum M(p_i)$ and thus

1) $\sum M(p_i') < \sum M(p_i)$.

This contradicts the assumption that $\|P|M\|$ is minimal for we can define $P' = P - \{p_i\}_j + \{p_i'\}_j$ and by 1) $\|P'|M\| < \|P|M\|$.

Thus C such that $\|C|M\| < \|C|T\|$ does not exist and by Ell there exists M' and σ such that $M[\sigma>M'$ and $M' \geq T$.

Since $M' \geq T$

$$\sum_i M'(\alpha_i) \geq \sum_i T(\alpha_i)$$

However $\sum_i T(\alpha_i) = \sum_i M(\alpha_i) + \sum_i M(p_i)$

Thus $\sum_i M'(\alpha_i) \geq \sum_i M(\alpha_i) + \sum_i M(p_i)$

But $\sum_i M'(\alpha_i) = \|E|M'\|$

$\sum_i M(\alpha_i) = \|E|M\|$

$\sum_i M(p_i) = \|P|M\|$

Thus $\|E|M'\| \geq \|E|M\| + \|P|M\|$

Since $\delta E + \delta P = 0$

$\|E|M\| + \|P|M\| = \|E|M'\| + \|P|M'\|$

Thus $\|E|M'\| \geq \|E|M'\| + \|P|M'\|$

Thus $0 \geq \|P|M'\|$

However $\|P|M\| \geq 0$ thus $\|P|M'\| = 0$

Since $U = -\delta P$

$\|\sigma|U\| = \|P|M\|$

Q.E.D.

The preceding theorem deals with expressions of the form

1) $\max_{M[\sigma\rangle} \|\sigma|U\|$ where σ ranges over the set of firing sequences which begin at M.

We may also evaluate expressions like

2) $\max_{M,M[\sigma\rangle} \|\sigma|U\|$ where M is assumed to range over the markings of some live marking class $\overleftrightarrow{M}$ and σ over the set of firing sequences beginning at M. An example of the distinction between 1) and 2) is the difference between the D_M and the D measure:

$D_M(x|y) = \max_{M[\sigma\rangle} \|\sigma|x-y\|$ $\qquad$ $D(x|y) = \max_{M,M[\sigma\rangle} \|\sigma|x-y\|$

Theorem G4

In a strongly connected live graph g

Let U be a linear expression in vertices such that there exists an expression E in arcs with positive coefficients such that $-\delta E = U$.

$$\max_{M, M[\sigma\rangle} \|\sigma|U\| = \min_{-\delta E = U = E'} \|E + E'|M\|$$

where E' is an expression in arcs with positive coefficients.

Proof: First we will show that E' exists. Since g is strongly connected there exists a circuit c which contains each arc at least once. This circuit defines a linear expression $C = \|a_1|c\|a_1 + \|a_2|c\|a_2 \ldots \|a_n|c\|a_n$ in arcs such that $\delta C = 0$.

Let N be a positive integer such that for all a_i

1) $N\|a_i|c\| > \|a_i|E\|$.

Then let $E' = NC - E$. By 1), E' has positive coefficients. Since $\delta E' = \delta NC - \delta E$ and $\delta NC = 0$,

$\delta E' = -\delta E$ thus $\delta E' = U$.

Now we will show that $\max_{M,M[\sigma\rangle} \|\sigma|U\| \leq \min_{-\delta E=U=\delta E'} \|E+E'|M\|$

Let σ be any sequence beginning at some $M \in \overset{\leftrightarrow}{M}$ such that $M[\sigma\rangle M'$.

Since $-\delta E = U$

2) $\|\sigma|U\| = \|E|M\| - \|E|M'\|$

However since E' has positive coefficients

$\|E'|M\| \geq 0$ thus

3) $\|E|M\| \leq \|E + E'|M\|$.

Furthermore $\|E|M'\| \geq 0$ thus

4) $-\|E|M'\| \leq 0$

Adding 4) and 3) we have $\|E|M\| - \|E|M'\| \leq \|E + E'|M\|$ thus by 2) $\|\sigma|U\| \leq \|E + E'|M\|$ for all $M,M[\sigma\rangle$, E and E' .

(Note again that $\|E + E'|M\|$ is independent of M ranging over $\overset{\leftrightarrow}{M}$ because $(E+E') = 0$.)

Thus in particular,

$$\max_{M,M[\sigma\rangle} \|\sigma|U\| \leq \min_{-\delta E=U=\delta E'} \|E + E'|M\|$$

Hence we need only show that there exist $M,M[\sigma\rangle$, E and E' such that $\|\sigma|U\| = \|E + E'|M\|$.

Let M_o be any marking in $\overset{\leftrightarrow}{M}$. By G3

$$\max_{M_o[\sigma\rangle} \|\sigma|-U\| = \min_{\delta E=U} \|E|M\|$$

Let σ_o be the maximal σ and E_o be the minimal E . Then if $M_o[\sigma_o\rangle M_1$, $\|E_o|M_1\| = 0$.

However

5) $$\max_{M_1[\sigma\rangle} \|\sigma|U\| = \min_{-\delta E=U} \|E|M_1\|$$

Let σ_1 be the maximal σ and E_1 be the minimal E. Clearly $-\delta E_1 = \delta E_o$. Since $\|E_o|M_1\| = 0$ $\|E_o + E_1|M_1\| = \|E_1|M_1\|$. By 5) $\|\sigma_1|U\| = \|E_1|M_1\|$ thus there exists M_1, σ_1, E_1 and E_o such that $-\delta E_1 = U = \delta E_o$ and $\|\sigma_1|U\| = \|E_1 + E_o|M\|$.

Thus $\max_{M,M[\sigma\rangle} \|\sigma|U\| = \min_{-\delta E=U=\delta E'} \|E + E'|M\|$

Q.E.D.

The preceding theorems enable us to generalize the measure $D(x|y)$ for single vertices x and y to a similar measure for sets.

Recall that

$$D_M(x|y) = \max_{M[\sigma\rangle} \|\sigma|x-y\|$$

$$D(x\ y) = \max_{M,M[\sigma\rangle} \|\sigma|x-y\|$$

Definition:

Let U and V be sets of vertices

$$D_M(U|V) = \max_{M[\sigma\rangle} \|\sigma|U-V\|$$

$$D(U|V) = \max_{M,M[\sigma\rangle} \|\sigma|U-V\|$$

Note that a set S of vertices may be expressed as a linear expression which places coefficient 1 on $v \in S$ and 0 on $v \notin S$. The reader will be pleased to discover that if x and y are single vertices

$D_M(\{x\}|\{y\}) = D_M(x|y)$

also $D(\{x\}|\{y\}) = D(x|y)$.

Theorem G5:

If $\mathcal{N}$ is live and strongly connected and U_1 , U_2 , and U_3 are vertex sets containing the same number of vertices; that is, $\|U_1|V\| = \|U_2|V\| = \|U_3|V\|$ then for all U_1 , U_2 , U_3

a. $D(U_1|U_1) = 0$

b. $D(U_1|U_2) = D(U_2|U_1)$

c. $D(U_1|U_3) \leq D(U_1|U_2) + D(U_2|U_3)$

Proof:

a) $D(U_1|U_1) = \max_{M,M[\sigma\rangle} \|\sigma|U_1 - U_1\| = \|\sigma|0\| = 0$.

b) Since $\min_{-\delta E=U=\delta E'} \|E + E'|M\|$ is symmetric with respect to E and E',

$$\min_{-\delta E=U=\delta E'} \|E + E'|M\| = \min_{-\delta E_1=(-U)=\delta E_1'} \|E_1 + E_1'|M\| .$$

Thus in a strongly connected live graph

1) $$\max_{m,M[\sigma\rangle} \|\sigma|U\| = \max_{M,M[\sigma\rangle} \|\sigma|-U\| .$$

Now if U and U' are vertex sets containing the same number of vertices, E such that $\delta E = U - U'$ exists, thus by 1) $D(U|U') = \max_{M,M[\sigma\rangle} \|\sigma|U - U'\| = \max_{M,M[\sigma\rangle} \|\sigma|U' - U\| = D(U'|U)$.

This means that if two vertex sets are the same size, the D measure is symmetric.

c) If $D(U_1|U_2) = N = \min_{-\delta E_1=U_1-U_2=\delta E_1'} \|E_1 + E_1'|M\|$

and $D(U_2|U_3) = K = \min_{-\delta E_2=U_2-U_3=\delta E_2'} \|E_2 + E_2'|M\|$

Then there exists $E_3 = E_1 + E_2$ and $E_3' = E_1' + E_2'$ such that $-\delta E_3 = -\delta E_1 - \delta E_2 = (U_1-U_2) + (U_2-U_3) = U_1 - U_3$ and

$\delta E_3' = \delta E_1' + \delta E_2' = (U_1-U_2) + (U_2-U_3) = U_1 - U_3$ with $\|E_3 + E_3'\| = D(U_1|U_2) + D(U_2|U_3)$.

$D(U_1|U_3) = \min_{-\delta E=U_1-U_3=\delta E'} \|E + E'\| \leq \|E_3 + E_3'\| = D(U_1|U_2) + D(U_2|U_3)$

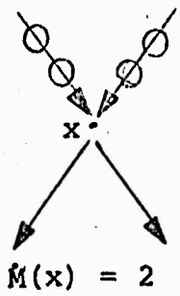
and

$D(U_1|U_3) \leq D(U_1|U_2) + D(U_2|U_3)$.

Q.E.D.

H. Set Firings

In the marked graph $\mathcal{G} = \langle g, M \rangle$, there may exist several vertices x such that $M[x\rangle$ exists. If $M[x\rangle$ exists and $M[y\rangle$ exists, x and y are said to be concurrently firable in M . Note: a vertex may be concurrently firable with itself.



The number of times x can be fired concurrently with itself at a given marking M is the minimum number of tokens on an input arc to x . We define the vertex marking of M , written $\dot{M}$, to be the function which assigns this number to each vertex.

$$\dot{M}(x) \triangleq \min_{\dot{\iota}(\alpha,x)} |\alpha|M|$$

A set firing is a vector in vertices which may be concurrently fired at a given marking M . We write $M[(\sigma)\rangle$ exists. Formally, if $\dot{\sigma}$ is a vector in vertices, $M[(\sigma)\rangle$ exists $\triangleq$ for all vertices x , $\|\sigma|x\| \leq \dot{M}(x)$. Given a marking M , the maximal set firing at M is a vector in vertices $\dot{\sigma}_M$ such that for all x

$$\|\sigma_M|x\| = \dot{M}(x) \ .$$

Let $\Sigma = (\sigma_1), (\sigma_2), (\sigma_3) \ldots (\sigma_n)$ be a sequence of set firings.

The length of Σ, written $|\Sigma|$ is defined as n.

$M_0[\Sigma\rangle M_n$ means $M_o[(\sigma_1)\rangle\ M_1[(\sigma_2)\rangle M_2\ldots[(\sigma_n)\rangle M_n$.

It is assumed that $M_{i-1}[(\sigma_i)\rangle$ exists for $0 \leq i \leq n$.

If $M[\Sigma\rangle M$, then Σ fires every vertex the same number of times. We will call this number K_Σ.

Theorem H1: Given a marked graph $\langle g,M\rangle$

Let $\Sigma = (\sigma_1),(\sigma_2),(\sigma_3)\ldots(\sigma_n) \neq 0$ and let $M[\Sigma\rangle M$.

Let C be any circuit in g.

Then $\dfrac{|\Sigma|}{K_\Sigma} \geq \dfrac{\|C|V\|}{\|C|M\|}$

Proof: Let σ be any set firing in Σ. Let V_c be the vector in vertices such that $\|x|V_c\| = \|x|C\|$ for all x.

Then clearly $\|\sigma_M|V_c\| \leq \|C|M\|$.

Thus $\sum_{i=1}^{|\Sigma|} \|\sigma_i|V_c\| \leq |\Sigma| \cdot \|C|M\|$

However $\sum_{i=1}^{|\Sigma|} \|\sigma_i|V_c\| = K_\Sigma \cdot \|V_c|V\| = K_\Sigma \cdot \|C|V\|$

Thus $K_\Sigma \cdot \|C|A\| \leq |\Sigma| \cdot \|C|M\|$ and $\dfrac{|\Sigma|}{K_\Sigma} \geq \dfrac{\|C|V\|}{\|C|M\|}$

Definition:

$i(m) \triangleq i$ modulo m

$i(m)$ is the least integer n such that

$i = n + K \cdot m$ where K is an integer.

Theorem H2: Let $\Sigma = (\sigma_0),(\sigma_1)\ldots(\sigma_{m-1}) \neq 0$ be any sequence of maximal set firings such that $M_0[(\sigma_0)\rangle M_1[(\sigma_1)\rangle\ldots M_{m-1}[(\sigma_{m-1})\rangle M_0$; that is, $M_0[\Sigma\rangle M_0$. Then there exists a circuit C such that

$$\frac{|\Sigma|}{K_\Sigma} = \frac{\|C|V\|}{\|C|M\|} .$$

Proof: First we will show that there exists a circuit $\overrightarrow{C\|x_0\alpha_0x_1\alpha_1\ldots x_{n-1}\alpha_{n-1}x_0}$ such that:

For all $i \geq 0$ there exists a set firing S_i in Σ such that

1a. $S_i = \sigma_{i(m)}$

2a. $x_{i(n)} \in S_i$

3a. $\|S_i|x_{i+1(n)}\| = \|\alpha_{i(n)}|M_{i(m)}\|$

We will now recursively define a backwards directed path $\overleftarrow{P\|y_0\alpha_0y_1\alpha_1\ldots}$ which contains such a circuit. With every $i \geq 0$ we will associate a set firing S'_i in Σ such that

1b. $S'_i = \sigma_j \Longrightarrow S'_{i+1} = \sigma_{j-1(m)}$

2b. $y_i \in S'_i$

3b. If $S'_i = \sigma_j$, $\|S'_i|y_{i-1}\| = \|\alpha_{i-1}|M_j\|$

Let y_0 be any vertex.

Let S'_0 be any set firing in Σ which contains y_0 .

Now assume $y_i \in S_i = \sigma_j$. Since $\sigma_{j-1(m)}$ is a maximal set firing,

$$\|\sigma_{j-1(m)}|y_i\| = \min_{\dot{\uparrow}(a,y_i)} \|a|M_{j-1(m)}\|$$

We define α_i as any arc a for which $\|a|M_{j-1(m)}\|$ minimal; i.e. $\|\alpha_i|M_{j-1(m)}\| = \|\sigma_{j-1(m)}|y_i\|$

Note that

$$\|\alpha_i|M_j\| = \|\alpha_i|M_{j-1(m)}\| - \|\sigma_{j-1(m)}|\bar{y}_i\| + \|\sigma_{j-1(m)}|\dot{\uparrow}(\alpha_i)\| =$$

$$\|\sigma_{j-1(m)}|\dot{\uparrow}(\alpha_i)\|$$

4b. $$\|\alpha_i|M_{j-1(m)}\| = \|\sigma_{j-1(m)}|\dot{\uparrow}(\alpha_i)\|$$

However since $\bar{y}_i \in S_i = \sigma_j$, $\|\alpha_i|M_j\|$, thus $\|\sigma_{j-1(m)}|\dot{\uparrow}(\alpha_i)\| \geq 1$; hence $\dot{\uparrow}(\alpha_i) \in \sigma_{j-1(m)}$. It is therefore not surprising that we define S'_{i+1} as $\sigma_{j-1(m)}$ and $\bar{y}_{i+1}$ as $\dot{\uparrow}(\alpha_i)$.

Now consider the set of ordered pairs $\langle \bar{y}_i, S'_i \rangle$. Since the number of vertices is finite and the number of set firings in Σ is finite, the number of such ordered pairs is finite. Since we have associated such an ordered pair with each vertex of P, there must exist integers j and k, $k > j$ such that $\langle \bar{y}_j, S'_j \rangle = \langle \bar{y}_n, S'_k \rangle$ i.e., y_i and y_k are the same vertex; S'_j and S'_k are the same set firing.

Consider the path $\overleftarrow{P' \| y_j \alpha_j \ldots \alpha_{k-1} y_k}$, $P' \subseteq P$.

Define $C' = \overrightarrow{C' \| y_k \alpha_{k-1} \cdots \alpha_j y_j}$.

Since $S'_i = \sigma_j \Longrightarrow S'_{i+1} = \sigma_{j-1(m)}$ and since $S'_k = S'_j$, there must exist an integer $j \leq k' \leq k$ such that $S'_{k'} = \sigma_0$. We define $\overrightarrow{C \| x_0 \alpha_0 x_1 \alpha_1 \cdots x_{n-1} \alpha_{n-1} x_0}$ as $\overrightarrow{C \| y_{k'}, \alpha_{k'-1}, \cdots \alpha_j y_k \alpha_{k-1} \cdots \alpha_{k'} y_{k'}}$. Correspondingly $S_0 \triangleq S'_{k'}$; if $S_i = S'_j$, $S_{i+1} \triangleq S'_{j+1}$.

It follows from 1b, 2b and 3b that C as defined satisfies properties 1a, 2a and 3a. It follows from 3a and the argument 4b that

4a. $\| S_i | X_{i(n)} \| = \| \alpha_i | M_{i+1(m)} \|$

We now define a family of paths

$$P_0 \triangleq \alpha_0$$

$$P_{i+1} \triangleq P_i + \alpha_{i+1}$$

Obviously $P_{n-1} = C$.

Definition:

$$\Sigma_0 \triangleq S_0$$

$$\Sigma_{i+1} \triangleq \Sigma_i + S_{i+1}$$

Since $S_i = \sigma_{i(m)}$ $i(m) = m - 1$ implies that $\Sigma_{m-1} = \Sigma$. Furthermore, since $S_n = S_0$ $\sigma_{n(m)} = \sigma_{0(m)}$ thus n is divisible by m . In fact, $\Sigma_{n-1} = \frac{n}{m} \cdot \Sigma_{m-1} = \frac{n}{m} \cdot \Sigma$ and $M_0 [\Sigma_i \rangle M_{i+1}$.

We will show by induction that

1c. $\| P_i | M_{i+1} \| = \| \Sigma_i | x_0 \|$ for $0 \leq i \leq n-1$.

For $i = 0$ we note that by property 4a

$$\|\Sigma_0|x_0\| = \|\sigma_0|x_0\| = |\alpha_0|M_1\| = \|P_0|M_1\|$$

Now assume 1c for i (Note: all M and σ subscripts are modulo m)

$$\|P_{i+1}|M_{i+2}\| = \|P_i|M_{i+2}\| + \|\alpha_{i+1}|M_{i+2}\| =$$
$$\left(\|P_i|M_{i+1}\| + \|\sigma_{i+1}|x_0\| - \|\sigma_{i+1}|x_{i+1}\|\right) + \|\alpha_{i+1}|M_{i+2}\| =$$
$$\left(\|P_i|M_{i+1}\| + \|\sigma_{i+1}|x_0\| - \|\sigma_{i+1}|x_{i+1}\|\right) +$$
$$\left(\|\alpha_{i+1}|M_{i+1}\| + \|\sigma_{i+1}|x_{i+1}\| - \|\sigma_{i+1}|x_{i+2}\|\right) =$$
$$\left(\|P_i|M_{i+1}\| + \|\sigma_{i+1}|x_0\|\right) + \left(\|\alpha_{i+1}|M_{i+1}\| - \|\sigma_{i+1}|x_{i+2}\|\right)$$

However, by inductive assumption, $\|P_i|M_{i+1}\| = \|\Sigma_i|x_0\|$, thus $\|P_i|M_{i+1}\| + \|\sigma_{i+1}|x_0\| = \|\Sigma_{i+1}|x_0\|$.

Thus we have

1d. $\|P_{i+1}|M_{i+2}\| = \|\Sigma_{i+1}|x_0\| + \left(\|\alpha_{i+1}|M_{i+1}\| - \|\sigma_{i+1}|x_{i+2}\|\right)$

By 3a, however $\|\sigma_{i+1}|x_{i+2}\| = |\alpha_{i+1}|M_{i+1}\| = \|\alpha_{i+1}|M_{i+1}\|$.

Hence the term in parenthesis in 1d is equal to zero and we have

3c. $\|P_{i+1}|M_{i+2}\| = \|\Sigma_{i+1}|x_0\|$.

Now, recall that $P_{n-1} = C$ and $\Sigma_{n-1} = \frac{n}{m}\Sigma$.

Since P_{n-1} is a circuit we may replace $\|P_{n-1}|M_n\|$ by $\|P_{n-1}|M\|$. By 3c, $\|P_{n-1}|M\| = \|\Sigma_{n-1}|x_0\| = \frac{n}{m}\|\Sigma|x_0\|$ that is,

2d. $\|C|M\| = \frac{n}{m}\|\Sigma|x_0\|$.

However:

$\|\Sigma|x_0\| = K_\Sigma$

$n = \|C|V\|$

$m = |\Sigma|$

Thus we can rewrite 2d:

$$|C|M|| = \frac{||C|V|| \cdot K_\Sigma}{|\Sigma|}$$

$$\frac{||C|M||}{||C|V||} = \frac{K_\Sigma}{|\Sigma|}$$

$$\frac{||C|V||}{||C|M||} = \frac{|\Sigma|}{K_\Sigma}$$

Q.E.D.

Theorem H3: In a finite, strongly connected marked graph $mg = \langle g, M \rangle$ let Σ be an infinite sequence of maximal set firings. Then Σ becomes periodic. That is, $\Sigma = \Sigma_1 + \overline{\Sigma}_2$ where $\overline{\Sigma}_2 = \Sigma_2\Sigma_2\Sigma_2\cdots$

Proof: This follows because

1. Σ is infinite.
2. Since g is finite and strongly connected, the number of different markings in $\vec{M}$ is finite.
3. For every M there exists a unique maximal set firing σ_M and hence a unique successor $M[\sigma_M\rangle$.

Theorem H4: In a finite strongly connected marked graph, let $\Sigma \neq 0$ be any set firing sequence and M any marking such that $M[\Sigma\rangle M$. Let C be any circuit.

$$\min_{M,\Sigma} \frac{|\Sigma|}{K_\Sigma} = \max_{C} \frac{||C|V||}{||C|M||}$$

Proof: By H1,

$\frac{|\Sigma|}{K_\Sigma} \geq \frac{||C|V||}{||C|M||}$ thus we need only show that there exists

an M and a Σ such that $M[\Sigma\rangle M$ and a C such that

$$\frac{|\Sigma|}{K_\Sigma} = \frac{\|C|V\|}{\|C|M\|} .$$

Since g is strongly connected, by H3, any infinite set firing sequence becomes periodic, thus we may pick an infinite sequence $\Sigma = \Sigma_1 + \overline{\Sigma}_2$. Obviously there exists an M' such that $M'[\Sigma_2\rangle M'$. By H2 there exists a circuit

$$\frac{|\Sigma_2|}{K_{\Sigma_2}} = \frac{\|C|V\|}{\|C|M\|} .$$

Theorem H5: Let Σ and Σ' be two sequences of maximal set firings such that $M[\Sigma\rangle M$ and $M'[\Sigma\rangle M'$. Then

$$\frac{|\Sigma|}{K_\Sigma} = \frac{|\Sigma'|}{K_{\Sigma'}} .$$

Proof: By H2 and H4, $\frac{|\Sigma|}{K_\Sigma} = \max_C \frac{\|C|V\|}{\|C|M\|} = \frac{|\Sigma'|}{K_{\Sigma'}}$.

H6. Interpretation of Maximal Set Firings.

Assume each vertex represents an event which requires one time unit. Then all the events in a set firing may occur concurrently. Thus if Σ is a set firing sequence, the number of time units required for the corresponding sequence of events is $|\Sigma|$.

If $M[\Sigma\rangle M$ then $\frac{|\Sigma|}{K_\Sigma}$ is a measure of the average number of time units per vertex firing. We call this measure wavelength and denote it by $\lambda(\Sigma)$. The number $\frac{1}{\lambda(\Sigma)}$ is expressed in vertex firings per time unit. We call

this measure frequency or throughput rate and denote it by $f(\Sigma)$.

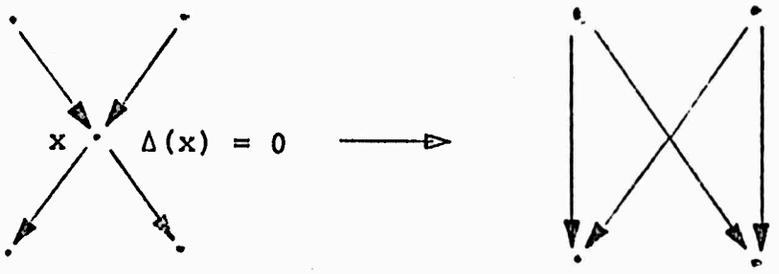
Thus theorem H4 states that the maximum average frequency of a graph is equal to the minimum ratio $\frac{|C|M|}{|C|V|}$.

Theorem H2 states that for any sequence Σ of maximal set firings, $f(\Sigma)$ is maximum. We may wish to assume that some events take many time units and some events take no time at all.

We can generalize our theorems to this case by the following transformation.

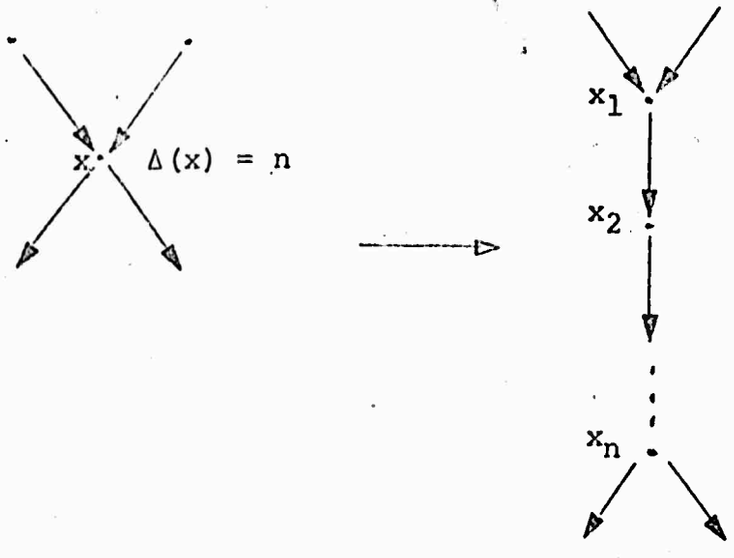
1. Assign to each vertex an integer $\Delta(x) \geq 0$.
2. If $\Delta(x) = 1$, leave x unaltered.
   If $\Delta(x) = 0$, replace x by a complete graph from the vertices which point to x to the vertices to which x points.

Example:



3. If $\Delta(x) > 1$, replace x by a string of vertices of length $\Delta(x)$.

Example:

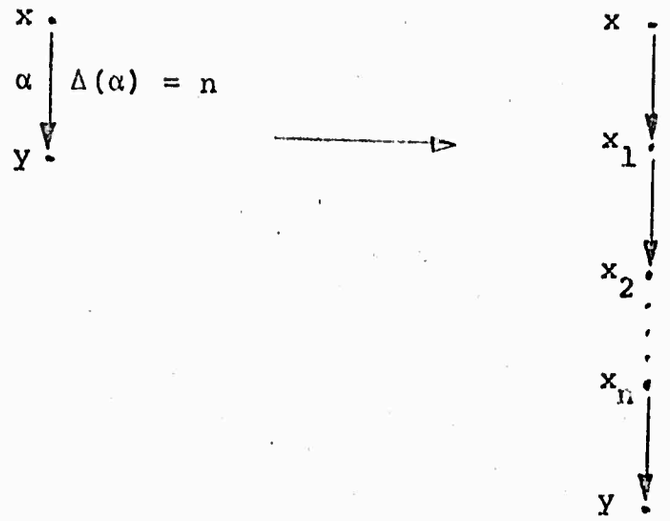


With this transformation complete, we can state:

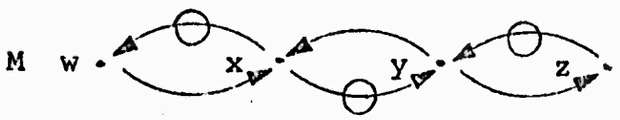
$$\max_{\Sigma} f(\Sigma) = \min_{C} \frac{\|C|M\|}{\|C|\Delta\|}$$ where Σ is any set firing sequence which begins and ends with the same marking. Furthermore, for any sequence Σ of maximal set firings which begins and ends at the same marking, $f(\Sigma)$ is maximum.

Note: Other timing constraints may be realized by graph transformations. For example, we can ensure that $\dot{M}(x) \leq K$ by placing a self loop on x with K tokens. Since this self loop L has one vertex, $\frac{\|L|M\|}{\|L|V\|} = K$; this implies that $f(\Sigma) \geq K$.

A time delay $\Delta(\alpha)$ may be placed on an arc by adding $\Delta(\alpha)$ vertices to it:



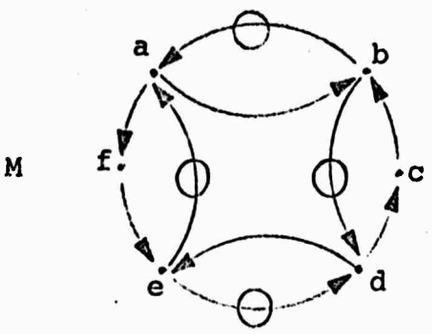
Examples:



$M[(w\ y),(x\ z)\rangle M$

$$\min_C \frac{|C|M|}{|C|V|} = \frac{1}{2}$$

$$f(\Sigma) = \frac{K_\Sigma}{|\Sigma|} = \frac{1}{2}$$



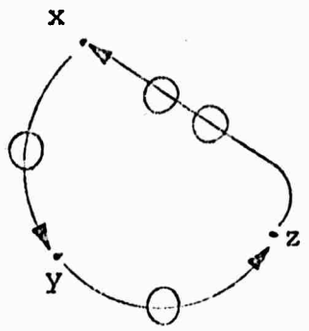
$$\min_C \frac{|C|M|}{|C|V|} = \frac{2}{6} = \frac{1}{3}$$

$\Sigma = (d\ a)(f\ c)(b\ c)$

$M[\Sigma\rangle M$

$$f(\Sigma) = \frac{1}{3}$$

M



$$\min_{C} \frac{\|C|M\|}{\|C|V\|} = \frac{4}{3}$$

$$\Sigma = (x,x,y,z)(y,y,z,x)(z,z,x,y) \qquad f(\Sigma) = \frac{4}{3}$$

STATE MACHINES AND INFORMATION

A. Introduction

A1. Scope

This paper introduces a concept of information flow through a system and provides an analysis of this flow for the case of finite state machines. This concept allows information to be identified, and measured in a manner consistent with existing measures of information. It allows us to trace the history of a quantum of information as it moves through its system environment.

A2. Information

A piece of information is usually thought of as an answer to a question. In regard to systems, information answers questions of the form: which of several possible next behaviors will the system exhibit? That is, the information resolves choices among possible next behaviors. More specifically, if the system is represented as a state machine, then it is resolution of choice among possible next state transitions. Such resolutions of choice will be called information input to a system.

We can say that a present system state contains the information which was earlier supplied to the system if the present state could only have been achieved in the presence of those choice resolutions. Then this past information can be "deduced" from the present state.

Finally, the system may pass to new states and thus

"forget" some old information. In other words, the new state is compatible with several possible choice resolutions on some earlier occasion. We will identify such occasions of forgetting with information output.

Here is an example -- a new coat of paint is to go on my wall. I supply the information red. One year later, the state of the wall still contains that information. Two years later, even though the color faded and is now a different red, it still contains that information. Three years later a new coat of paint is to go on the wall again. When it does -- and whether it is red or not -- the state of the wall no longer "remembers" my original decision.

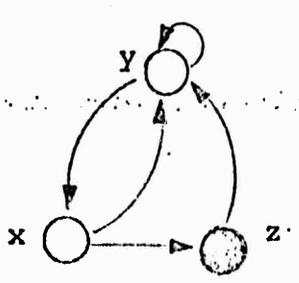
Another example: we may imagine the system as a library and the information contained in the system as the books inside.

Input and output correspond to the movement of books into and out of the library. The contents of a book are remembered by the library as long as it retains the book and forgotten when the book leaves.

A3. State Machines

We will represent a state machine by a directed graph. The vertices of the graph represent the states of the machine and the arcs represent possible state transitions.

A state machine graph



We will not assume that the state machine graph is accompanied by an alphabet of input-output characters or a table of state transitions. Rather, we will attempt to construct a set of fundamental information quanta appropriate to the state machine based only on its graph structure. We may indicate the state of the machine by placing a token on one of its vertices (vertex z above).

A4. Input

If a machine is in a state from which it is possible to transit to several states, (e.g. states x and y above) it requires an input of information to determine which transition is to take place. It is with reference to the flow of this type of information that we wish to analyze the behavior of the machine.

We therefore assume that the input to a state machine is exactly equivalent to the choice of the next state from the set of possible successors.

A5. Output

If a machine arrives at a state which may be reached from several different states (e.g. y above) it "forgets" which arc the state was entered by. We will assume that the information which is lost on arrival at a state is exactly equivalent to the specification of the arc by which the state was entered. That is, what the machine forgets when it arrives at a state is exactly the information which would be required to back up correctly to the

immediately preceding state.

We will also assume that the information content of the machine is a function of its state and that information which is output is composed of information which has been input.

A6. Information Content

Imagine a state machine S in state x (Fig. 1.) which receives an input which causes it to change to state y . We may think of this input as simply the name of the arc α from x to y .

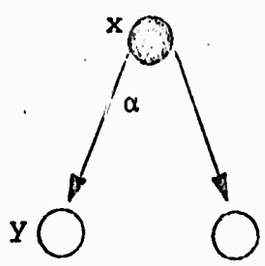


Fig. 1.

Let us imagine that we pause before arriving at y at an intermediate state placed on the arc α (Fig. 2.).

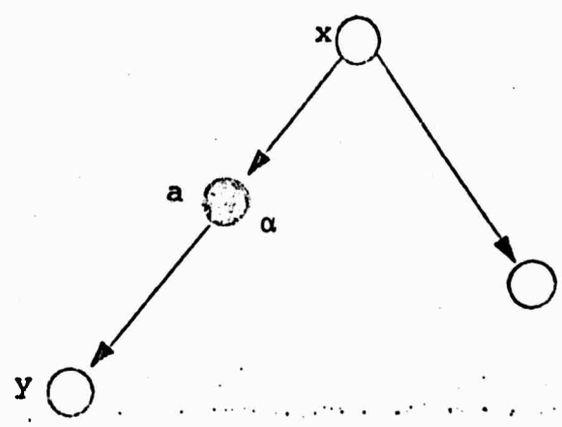


Fig. 2.

When S is in this state (a) we may infer that the last input at the state x was α . Thus we say that the information represented by the input α is contained in S at the state a .

Now we continue to state y . It is possible that we can now no longer infer that the last input at x was α . For example, there may be a path from x to y which does not go through α . In this case we say that some part of the information which was present at a is lost on arrival at the state y .

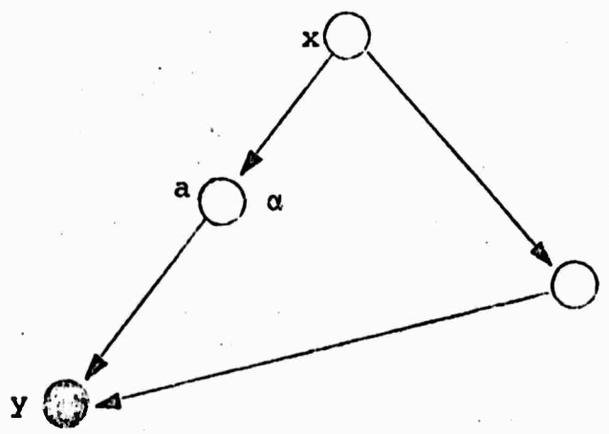
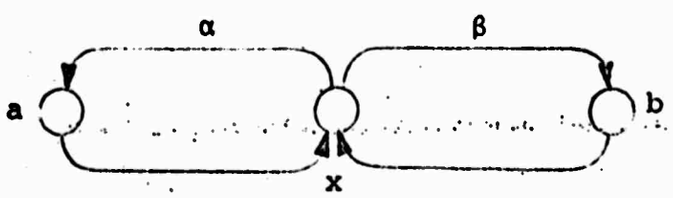


Fig. 3.

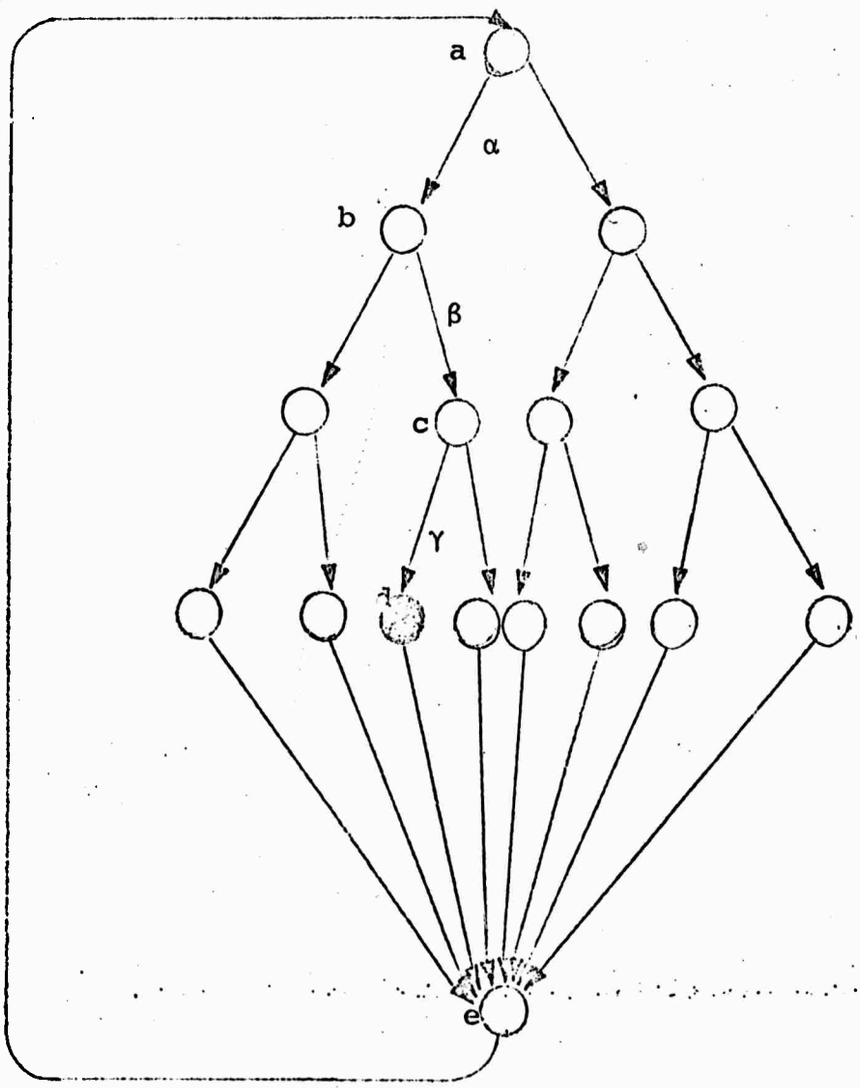
Since we have assumed that output information is composed of old input information, we wish to determine exactly what the relationship is between outputs and the past inputs of which they are composed.

EXAMPLE:



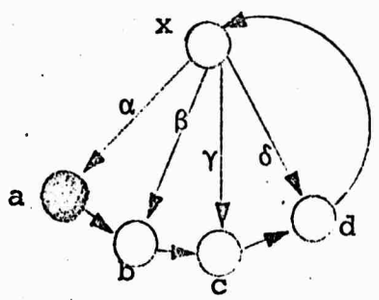
At the state x we may input α and change the machine to state a. At state a the machine <u>remembers</u> the input α; that is, the only way to reach state a is to leave x by α. Now the machine changes from a to x. The information which was contained in the machine at a is output; we no longer know which state transition occurred last time we were at state x.

It is possible to store a larger amount of information in a state machine of the following type:



If we start at state a and move to d through states b and c, we have stored in the machine the information corresponding to the choices of the arcs α, β and γ. When we move from d to e all of this information is lost.

Information may be input in a large quantity and output piece by piece. If we start at x and input α, this information is retained by the machine at state a.



That is, we can deduce that the last input at x was α. If we now move to state b, however, part of this information is lost. We are no longer sure whether the last input at x was α or β. We have not lost all of the information the input α represents, however, because we can still say that the last input at x was neither γ nor δ. Specifically, the arc α was a selection of one arc out of four possibilities. The selection of one of the two arcs α and β out of these four possibilities is still recorded in the state of the machine, but the selection of α out of the pair α, β was output on

arrival at b . If we now continue to c and d , the rest of the information represented by the input α will be lost.

The preceding example suggests that if we are to find a set of fundamental quanta with which to express information flow through the state machine, we must allow an input to consist of many such quanta so that it may be output piece by piece.

A7. Recent Inputs

If the information content of the machine S is to be a function of its state, any change in information content which occurs after the machine leaves a state x must vanish by the time the machine returns there. Any information which is output must be re-input, and any information which is input must be output.

It therefore follows that if information is output by the machine, it can only be composed of some combination of the most recent inputs at each state. This is because any earlier input at a state x would have been output by the time the machine returned to x . We will now investigate the properties of the set of most recent inputs of a state machine. This investigation will enable us to formalize the intuitive connection between input and output suggested by the preceding examples.

B. Decision Graphs

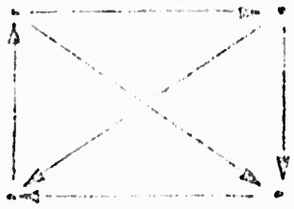
B1. We shall restrict our attention to state machines which can be represented by finite, strongly connected graphs where the vertices represent the states and the arcs represent the transitions.

B2. We can think of the history of a state machine S as a path σ in the corresponding graph. We may either wish to draw attention to the state sequence (i.e. vertex sequence) which the path defines, or to the firing sequence (i.e. arc sequence). In either case, we will refer to the path σ and let context make clear which sequence is of interest.

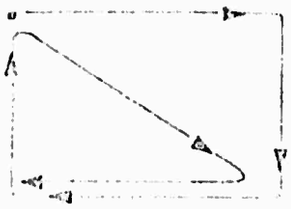
B3. Let P be a finite directed path in a graph S. We can associate with every vertex x on the path P (except, perhaps, the terminal vertex) an arc, called the last exit from x, namely the latest arc on P incident out of x. Then we can call an arc of P a last exit if it is the last exit of some vertex. The last exits of P represent the latest decisions which were made at the vertices of P. A Decision-graph D or simply D-graph is the set of last exits (E) of P, plus the vertices of these arcs with the terminal vertex of P (ω) distinguished. ω is called the root of the D-graph. We write $D = \langle E, \omega \rangle$.

Examples

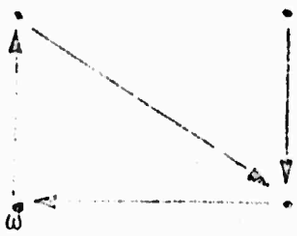
(1)



S

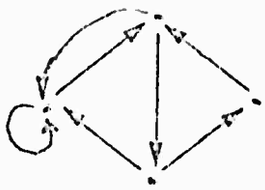


P

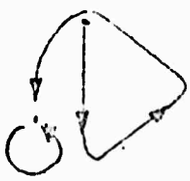


D

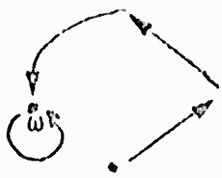
(2)



S

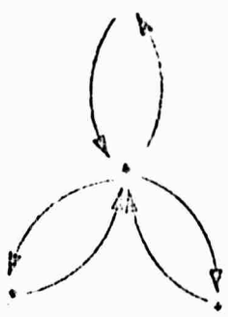


P

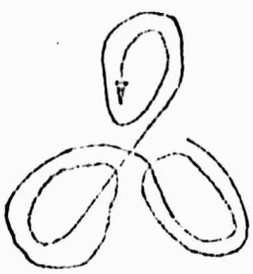


D

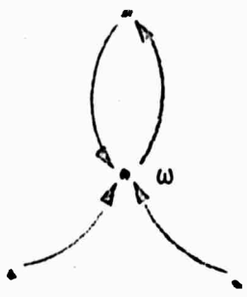
(3)



S



P



D

We will call a path in S long if it comes to and exits from every vertex of S .

The D-graphs of S are the D-graphs of all long paths of S .

B4. Let P' be a terminal segment of the path P . Then the D-graph of P' must be a subgraph of the D-graph of P .

Proof: This follows because, if a vertex lies on P' as well as P then its last exit in P' must be the same as its last exit in P .

B5. Let $\overrightarrow{P\|...x}$ be a path and D its D-graph. Then, in D , there is a path from every vertex on P to the vertex x , the root of D .

Proof: This is easy to see with an induction on the number of arcs in P . This statement is obvious if P is a single arc.

Now consider a path, Q , with n + 1 arcs ($n \geq 1$) . Suppose that the k^{th} arc of Q is arc a from vertex y to vertex z , and that a is the first last exit of Q . If Q' is the terminal segment of Q which begins just after the k^{th} arc of Q , then its D-graph, D' is a subgraph of D , differing from D only in that it lacks vertex y and its exit arc.

By inductive hypothesis our assertion is true

for D' , and therefore obviously for D .

B6. If the path $\overrightarrow{P\|...x}$ never exists from x (i.e, is not of the form $\overrightarrow{\|...x...x}$) then the D-graph D of P must be a tree rooted at x .

Proof: By B5, there is a path from every vertex in D to x . Since each vertex has at most one output arc in D , these paths are unique. Thus any circuit in D would have to pass through x -- but x , by hypothesis has no exit arc in D .

Conversely, if P is of the form $\overrightarrow{\|...x...x}$ then D must contain a simple circuit, and in fact exactly one -- namely the simple circuit consisting of the last exit arc from x to some vertex y and the unique simple path in D from y to x .

We can now describe the D-graphs of a finite directed graph S . Since the generating paths are "long" (i.e., come into and out of every vertex of S) the resulting D-graph can be visualized as a maximal tree (directed toward the root), rooted at the terminal vertex of the path, together with one arc leaving the root and thus closing a circuit. For the rest of this discussion, D-graph means D-graph of a long path.

B7. Given a finite strongly directed graph S, we wish to show that any maximal directed tree together with one arc out of the root x is a D-graph of S with root x. In other words, given the maximal tree with root x and the additional arc one must construct a path terminating at x with that tree and that arc as its D-graph.

Proof: Let T be a maximal tree in S, x_o its root and α any arc leaving x_o. Now choose any path P_o which begins at x_o, exits x_o for the first time by the arc α, and contains every leaf node of T. This path surely exists since S is strongly connected.

Now let $P_o = \overrightarrow{P_o \| x_o, x_1, x_2 \ldots x_j}$.

Now define for $0 \leq i \leq j-1$

$P_o = P_o$

$P_i = \overrightarrow{P_i \| x_o, x_1, \ldots x_{j-i}}$

(P_i is simply P with the last i vertices removed.)

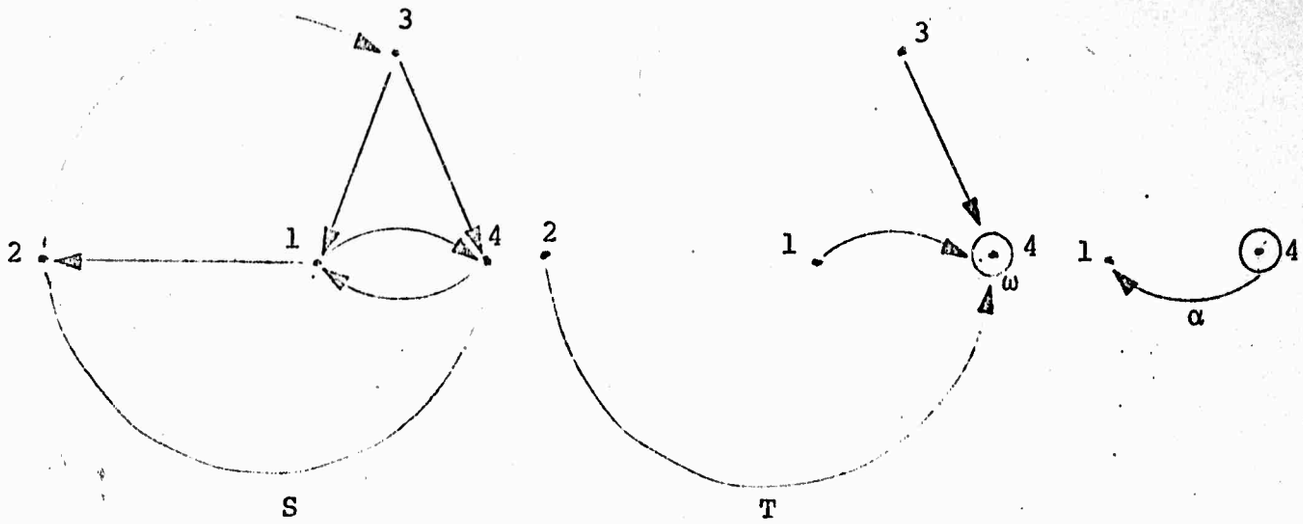
Now since T is maximal there exists a unique path Q in T from every vertex to x_o.

For $\overrightarrow{P_i \| x_o, x_1, \ldots x_{j-i}}$ define Q_i to be the unique path in T from x_{j-i} to x_o.

Now we can define the path P which yields a D-graph consisting of T and α rooted at x_o.

$P = P_o Q_o P_1 Q_1 \ldots P_{j-1} Q_{j-1}$

The following is an example of this construction:



P_o 4,1,2,3

P_1 4,1,2

P_2 4,1

P 4,1,2,3,4,1,2,4,1,4

Now we will show that in fact the D-graph D of P is $\langle T + \alpha, x_o \rangle$. P terminates at x_o, thus D is rooted at x_o. The last exit arc in P from x_o is the path P_{j-1}, which is simply the first arc α of P_o, thus $\alpha \in D$.

Because P covers every path from a leaf of T to

the root of T, it covers every arc of T. Since T contains every vertex, and since P begins with an exit from the root of the tree, P exits at least once from every vertex of S. Therefore P is long.

If a vertex does not lie on P then every exit from that vertex in P (particularly the last one) lies on a Q path and hence on the tree. If a vertex $y \neq x_o$ does lie on some path P_i then let P_k be the last such path in the order in which the paths are enumerated. Since P_k is the last path containing y, y must be the terminal vertex of P_k. Thus the last exit from y is on Q_k and hence on the tree.

B6 and B7 allow us to state the following theorems:

B8. $D = \langle A, \omega \rangle$ is a D-graph of S if and only if $A = T + \alpha$ where T is any maximal directed tree rooted at ω and α is any exit of ω.

B9. The number of different D-graphs rooted at a vertex x is equal to the product of the number of maximal trees with root x and the number of output arcs of x.

B10. Let $\overrightarrow{P|...y,x}$ be a long path in S , and D its D-graph. The arc entering x which lies on the unique circuit of D is the arc in P from y to x .

Proof: Consider the D-graph, D' , of the path $\overrightarrow{P'|...y}$ with P' identical to P less its terminal arc. D' differs from D in at most one respect: it may contain a different exit arc from y or possibly no exit arc from y . Since P is long, P' exits from x and hence D' contains a single path from x to y which exits unaltered in D . In D , the arc a is a path from y to x and therefore the arc must lie on the unique circuit of D .

B11. If S is interpreted as a state transition diagram and P as a state sequence for S then one can interpret the D-graph of P as the record of all last decisions made in generating P . The interpretation of B10 is then the following:

The records of all last decisions are sufficient to determine the last step of the state sequence (i.e. last arc of the path) and therefore sufficient to determine the next-to-last state. Put another way: the information for how to back up one step is contained in the record of last decisions.

Our initial assumptions stated that the information output upon arriving at a state is equivalent to the information required to back up correctly to the immediately

preceding state. We also assumed that an output is composed of old inputs, in fact of the most recent inputs. Since the information required to back up correctly is contained in the set of last decisions (Theorem B10) our two assumptions are consistent.

B 12. In B10 it was shown that the present D-graph of a state machine uniquely determines the arc by which the present state was entered. For a specific state however, less information than the entire D-graph may be sufficient to uniquely determine this arc (for example a state with only one input arc). In this section we wish to determine exactly what information about last decisions is required to determine the entrance to a given state.

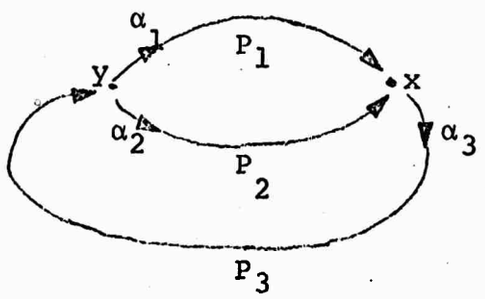
Definition: A partial D-graph of a set of vertices V is a D-graph less the last exits of vertices not belonging to the V set. We write $D(V) = \langle E, \omega \rangle$ for the partial D-graph of the set V rooted at ω where E is the set of last exits from V . (V may be null.)

Definition: Given a state x, $\overset{\nabla}{x}$ is a set of states such that:

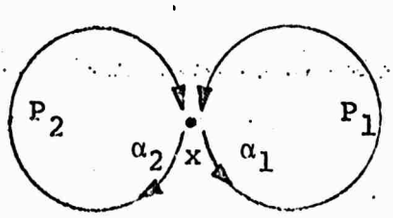
1. Any partial D-graph $D(\overset{\nabla}{x}) = \langle E,x \rangle$ uniquely determines the entry arc of x. More formally, for any partial D-graph $D(\overset{\nabla}{x}) = \langle E,x \rangle$, every D-graph $D = \langle A,x \rangle$, where $E \subseteq A$, has the same arc α entering x on its unique circuit.
2. $\overset{\nabla}{x}$ is minimal with respect to property 1.

B 13. Theorem: For a given state x, $\overset{\nabla}{x}$ is unique and consists of all vertices y with the following property:

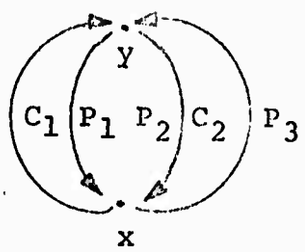
3. There exist two non-empty paths P_1 and P_2 from y to x which intersect only at x and y, and a path P_3 from x to y which intersects P_1 and P_2 only at x and y.



Note: P_3 may be null. This is the case if x itself satisfies property 3:



Proof: First we will show that any partial D-graph of the set of vertices V described in 3 uniquely determines the entry arc of x. To do this we will show that if $\langle A,x\rangle$ is a partial D-graph of V, there cannot exist D-graphs containing A whose circuits enter x by different arcs. Let C_1 and C_2 be two simple circuits which enter x through different arcs. Now traverse C_1 backwards from x. Let y be the first vertex encountered which is also on C_2.



Since y is the first such vertex, path $P_1 \subseteq C_1$ from y to x intersects the path $P_2 \subseteq C_2$ from y to x only at the vertices y and x. Furthermore, the path $P_3 = C_2 - P_2$ intersects P_1 and P_2 only at x and y. Thus y satisfies property 3 and hence $y \in V$. Therefore any partial D-graph $D(V) = \langle A,x\rangle$ contains a specific exit arc of y. Since C_1 and C_2 exit y by different arcs, C_1 and C_2 cannot both be contained in D-graphs which contain A.

Now we will show that every vertex y which satisfies property 3 must be in $\overset{\nabla}{x}$. To do this, we will show that there exists a partial D-graph of the set $S - \{y\}$, (where

S is the set of vertices of the graph) , $D(S-\{y\}) = \langle A,x \rangle$ which does not uniquely determine the entry arc of x . This would imply that no set of vertices not containing y could satisfy property 1 of $\overset{\nabla}{x}$.

Assume y satisfies property 3. Let α_1 , α_2 and α_3 be the first arcs, if any, of P_1 , P_2 , and P_3 . Consider the set of arcs $\{P_1 \cup P_2 \cup P_3\}$. Delete from this set all arcs leaving x and the first arc α_1 of P_1 . (See arcs in the above diagrams.)

The remaining set of arcs is a tree rooted at x . Add arcs to this set until it becomes a maximal tree T .

Now define

$$E \overset{\Delta}{=} T - \{\alpha_2\} + \{\alpha_3\} .$$

Then $\langle E,x \rangle$ is a partial D-graph of the set $S - \{y\}$. However, $\langle E+\alpha_1 , x \rangle$ is a D-graph whose unique circuit contains the last arc of P_1 ; $\langle E+\alpha_2 , x \rangle$ is a D-graph whose unique circuit contains the last arc of P_2 . Since these arcs both enter x and cannot be the same, $\langle E,x \rangle$ does not uniquely determine the entry arc of x . Thus $y \in \overset{\nabla}{x}$.

Q.E.D.

B 14. By definition $\overset{\nabla}{x}$ is a minimal set of vertices whose last exits are always sufficient to determine the entry arc of x . The specification of this arc represents the information which is output when x is

entered by the arc. The specification of this arc as a function of the past decisions or inputs at the vertices in $\overset{\nabla}{x}$ suggests that the information output at x is a function of the information input at the states in $\overset{\nabla}{x}$.

In this section we will further explore this connection.

Definition: If x is an arc or vertex, Cx is the set of simple circuits which contain x.

Note that Cx also specifies a class of D-graphs, namely all those D-graphs which contain circuits in Cx.

For each vertex x the information set Ix is defined as the complement of Cx. Thus a large information set means a large set of excluded circuits and hence a small set of included D-graphs; and therefore a low steady state probability.

For an arc α from x to y,

$$I\alpha \triangleq Cx - C\alpha$$

$$O\alpha \triangleq Cy - C\alpha$$

Thus in the state transition $x \cdot \xrightarrow{\alpha} \cdot y$

$$Iy = Ix + I\alpha - O\alpha$$

If a circuit c is an element of $I\alpha$, we say c is input at x in the state transition $x \cdot \xrightarrow{\alpha} \cdot y$. If c is an element of $O\alpha$, we say c is output at y in the state transition $x \cdot \xrightarrow{\alpha} \cdot y$.

If c is output at y and if c was last input at

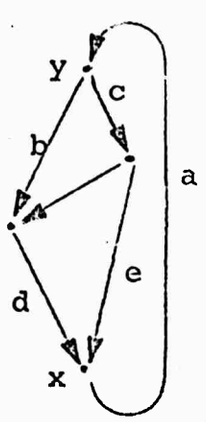
a vertex z, we say $c(z)$ is output at y.

B 15. Theorem: $y \in \overset{\nabla}{x}$ if and only if there exists a simple circuit c such that c may be input at y and $c(y)$ may be output at x.

→ If $y \in \overset{\nabla}{x}$ then by B 20 the paths P_1, P_2, and P_3 exist as defined in 3. Thus beginning at y we can move to x via P_1 and input the circuit $c = P_2 \cup P_3$ at y. Since $P_1 \cap c = \{x,y\}$, $c(y)$ is output at x.

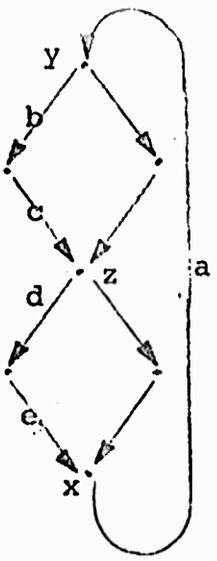
← Conversely if c may be input at y and $c(y)$ may be output at x, there exists a path P_1 from y to x such that $P_1 \cap c = \{x,y\}$. Now define P_2 and P_3 to be the paths in c from y to x and x to y respectively. The paths P_1, P_2 and P_3 satisfy requirement 3 for y. Thus by B 20, $y \in \overset{\nabla}{x}$.

Examples:



$y \in \overset{\nabla}{x}$

Both $c_1 = \{a,c,e\}$ and $c_2 = \{a,b,d\}$ may be input at y; $c_1(y)$ and $c_2(y)$ may be output at x.



$y \notin \overset{\nabla}{x}$

$C = \{a,b,c,d,e\}$ may be input at y and output at x, however $C(y)$ is output at z and $C(z)$ is output at x.

This means that the decision made at y is output at z and thus has no influence on the entry arc (and hence the output circuit set) at x.

B 16. With theorem B 15 in mind we define $\underset{\nabla}{x}$ to be the set of all states y such that there exists a curcuit c which may be input at x, such that $c(x)$ may be output at y.

Thus, while $\overset{\nabla}{x}$ is the set of states whose last inputs may influence the next output at x, $\underset{\nabla}{x}$ is the set of states whose next outputs may be influenced by the last input at x.

Hence $\underset{\nabla}{x} = \{y \mid x \epsilon \overset{\nabla}{y}\}$.

B 17. On the basis of Theorem B 15 we propose that the set of simple circuits of a state machine graph, with each circuit representing a class of D-graphs (those which contain the circuit) are a correct set of elementary information quanta to represent information flow in state machines.

The reader may verify that this set exhibits the

intuitive properties of input and output presented in the examples of section A.

C. D-graphs and Probability

Each simple circuit of a state machine graph may be thought of as a quantum of information. If we associate with a circuit C the set of D-graphs Δ_C which contain C, the information C represents may be stated thus: "The present D-graph is in Δ_C". If a piece of information consists of several such quanta C_1, C_2, C_3 we may say: "The present D-graph is in $\Delta_{C_1} \cup \Delta_{C_2} \cup \Delta_{C_3}$". The simplest method of measuring the amount of information represented by a set of circuits C_1, C_2, C_3 is to count the number of D-graphs in $\Delta_{C_1} \cup \Delta_{C_2} \cup \Delta_{C_3}$. In this section we will show that this method of information measurement is consistent with the existing measures of information in terms of probability. We will prove that the D-graphs of a state machine are equi-probable (Theorem C1). Thus the number of D-graphs in an information quantum is directly related to its probability.

When we say the present D-graph of S is D we assume that the present state of S is the terminal vertex of a long path P and the D-graph of P is D. The steady state probability of a D-graph $D = \langle A, \omega \rangle$ in

S is the probability that the present state of S is ω and A is the set of last exits from the vertices of S.

C1. Theorem: In the steady state of S, D-graphs are equiprobable.

Proof: From S we will construct a new state machine S' whose states are the D-graphs of S. Then we will show that an equiprobable distribution satisfies the steady-state equations of S'. Since S is represented by a strongly connected graph, the associated Markov process is irreducible. Thus these equations have a unique solution[1] and the theorem is proved.

Let S be a state machine at time t with state ω and present D-graph D_1. Clearly ω is the root of D_1. We write $P(D_1 \rightarrow D_2) = p$ if p is the probability that D_2 will be the D-graph of S at time $t + 1$.

Let the states of S' be the D-graphs of S. The transition probability in S' from state D_i to D_j is defined as $P(D_i \rightarrow D_j)$.

Now for any $D = \langle E, x \rangle$ the number of different D-graphs D' for which $P(D \rightarrow D') \neq 0$ is simply the number A_x of output arcs of x. Furthermore since we are assuming equiprobable exits in S, these A_x output arcs are equiprobable, hence if $P(D \rightarrow D') \neq 0$, $P(D \rightarrow D') = \frac{1}{A_x}$.

---

[1] Feller, William. An Introduction to Probability Theory and its Applications, Vol. I, 2nd. ed. New York: Wiley, 1957, p. 408.

By B10, the immediately preceding state of S is uniquely determined by the present D-graph of S. Thus, for any D , the D-graphs D_j for which $P(D_j \to D) \neq 0$ are all rooted at a unique vertex ω , which is simply the immediately preceding state determined by D .

Now let Δ be the set of all D_j such that $P(D_j \to D) \neq 0$.

If $D_j \in \Delta$, D_j differs from D only in respect to the last exit from ω . Thus $D \cap D_j$ always contains a tree T rooted at ω . Furthermore any D-graph rooted at ω which contains T is in Δ . Hence Δ is exactly the set of D-graphs rooted at ω which contain T . The number of such D-graphs is the number A_ω of output arcs of ω . That is, $|\Delta| = A_\omega$.

Now define $P(D_j)$ as the steady state probability of D_j in S' . Then the steady state equations for S' are:

For all D

$$P(D) = \sum_j P(D_j)\ P(D_j \to D)$$

If $P(D_j \to D) \neq 0$ $P(D_j \to D) = \frac{1}{A_\omega}$ where A_ω is the number of output arcs of the root of D_j. Furthermore every $D_j \in \Delta$ has the same root ω. Thus we may write:

$$P(D) = \sum_{D_j \in \Delta} P(D_j) \frac{1}{A_\omega}$$

Now assume all D-graph probabilities are equal and thus equal to $P(D)$. Then we have:

$$P(D) = \sum_{D_j \in \Delta} P(D) \frac{1}{A_\omega}$$

However, $|\Delta| = A_\omega$ thus

$$P(D) = \sum_{i=1}^{A_\omega} P(D) \cdot \frac{1}{A_\omega} = A_\omega \cdot P(D) \cdot \frac{1}{A_\omega} = P(D)$$

and we have $P(D) = P(D)$ which is always true.

Thus the steady state equations are satisfied if the D-graphs of S are equiprobable. Since their solution is unique, they are satisfied only if the D-graphs of S are equiprobable.

The preceding theorem states that D-graphs are equiprobable in steady state. Since S always has exactly one D-graph, $P(D) = \frac{1}{\Sigma}$ where Σ is the total number of D-graphs of S.

Results of theorem C1.

C2. Let D_x be the number of D-graphs rooted at x. The steady state probability of x is $\frac{D_x}{\Sigma}$. This

follows immediately from C1. since S is in state x if and only if its present D-graph is rooted at x .

C3. The probability of a D-graph D given that ω is the present state is 0 if ω is not the root of D and otherwise $\frac{1}{D_\omega}$ where D_ω is the number of D-graphs rooted at ω .

If ω is the root of D , $P(D) = P(\omega)\ P(D|\omega)$. However $P(D) = \frac{1}{\Sigma}$; $P(\omega) = \frac{D_\omega}{\Sigma}$ thus $\frac{1}{\Sigma} = \frac{D_\omega}{\Sigma} P(D|\omega)$ and $P(D|\omega) = \frac{1}{D_\omega}$.

C4. Let D_{Ax} be the number of D-graphs rooted at x which contain the set of arcs A . Given the present state x , the probability that the present D-graph includes some set of arcs A is $\frac{D_{Ax}}{D_x}$. $P(A|x) = \sum_D P(D|x)\ P(A|D)$. By C3. $P(D|x) = \frac{1}{D_x}$; $P(A|D)$ is 1 if $A \subset D$ and 0 if $A \not\subset D$.

C5. Let D_α be the number of D-graphs rooted at the vertex $\dot{\uparrow}(\alpha)$ and containing α on a circuit. (Note that D_α is also the number of D-graphs rooted at $\dot{\uparrow}(\alpha)$ and containing α on a circuit.) The steady state probability that an entrance to x is α where α is some arc leading into x is $\frac{D_\alpha}{D_x}$.

Let $P(\alpha x)$ be the steady state probability that x is entered through the arc α. By B10, if S is at state x, x was entered through α if and only if the present D-graph (rooted at x) is a member of a set containing α on a circuit. By C3, the probability of this set is $\frac{D_\alpha}{D_x}$.

C6. We can now establish the relationship between state transitions in a state machine and changes in the amount of information contained in the state machine.

Assume S is at state x and transits to state y via arc α. We will associate the exit from x with input and the arrival at y with output.

The amount of information input required to leave x by α is defined as $-\log_2 p(x\alpha)$ where $p(x\alpha)$ is the probability of leaving x by α.

The amount of information which is output upon arrival at y is $-\log p(\alpha y)$ where $p(\alpha y)$ is the probability of entering y through α. This is simply the amount of information which would be required to back up from y along α.

Thus if the initial information content at x is

$-\log I$, the information content $-\log(I')$ after the transition from x to y is

$$-\log I - \log p(x\alpha) + \log p(\alpha y) =$$

$$-\log I \frac{p(\alpha y)}{p(x\alpha)} \quad .$$

For a longer sequence of states and transitions $x_o\alpha_o x_1\alpha_1 \ldots x_n$ we have

$$-\log(I') = -\log\left(I \frac{p(\alpha_o x_1)}{p(x_o\alpha_o)} \cdot \frac{p(\alpha_1 x_2)}{p(x_1\alpha_1)} \cdots \frac{p(\alpha_{n-1}x_n)}{p(x_{n-1}\alpha_{n-1})}\right)$$

C7. Hereafter by information content we will mean the argument of the $-\log$ function rather than the value of that function.

If $x\alpha y$ is a state transition, $p(x\alpha) = \frac{1}{N}$ where N is the number of output arcs of x. By B8 and B9, $D_x = N \cdot T$ where T is the number of trees rooted at x. Furthermore, D_α, the number of D-graphs rooted at x and containing α on a circuit, is equal to T, thus $p(x\alpha) = \frac{1}{N} = \frac{D_\alpha}{D_x}$. Similarly (by B16) $p(\alpha y) = \frac{D_\alpha}{D_y}$. Thus

$$I' = I \frac{p(\alpha y)}{p(\alpha x)} = I \frac{D_\alpha}{D_y} \cdot \frac{D_x}{D_\alpha} = I \frac{D_x}{D_y} \quad .$$

For a longer sequence $x_o, x_1, x_2, \ldots x_n$

$$I' = I \frac{D_{x_o}}{D_{x_1}} \cdot \frac{D_{x_1}}{D_{x_2}} \cdot \frac{D_{x_2}}{D_{x_3}} \cdots \frac{D_{x_{n-1}}}{D_{x_n}} = I \frac{D_{x_o}}{D_{x_n}}$$

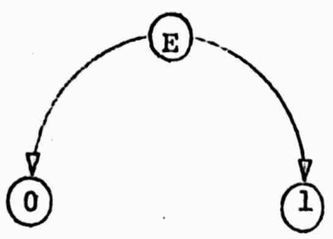
In particular for any sequence which is a circuit,

$D_{x_o} = D_{x_n}$ and $I' = I$. This measure allows us to speak of the information difference from state x to state y as $\frac{D_x}{D_y}$, which is independent of the path used in moving from x to y . Note that (by B13) $\frac{D_x}{D_y} = \frac{D_{x/\Sigma}}{D_{y/\Sigma}} = \frac{p(x)}{p(y)}$. Thus it is consistent to assume that at the state x , $I = \frac{1}{p(x)}$ where $p(x)$ is the steady state probability of x . For if we move from x to y ,

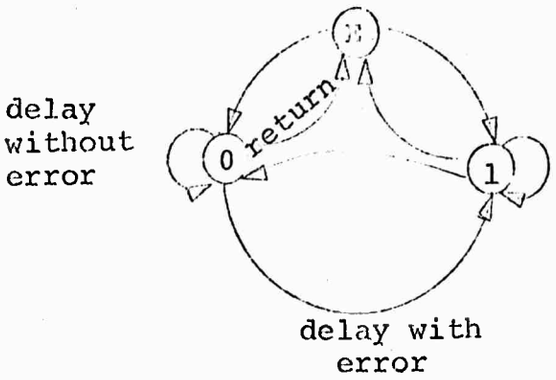
$$I' = I\frac{D_x}{D_y} = I\frac{p(x)}{p(y)} \quad \text{and} \quad \frac{I'}{I} = \frac{1/p(y)}{1/p(x)} \ .$$

D. An Interpreted Analysis

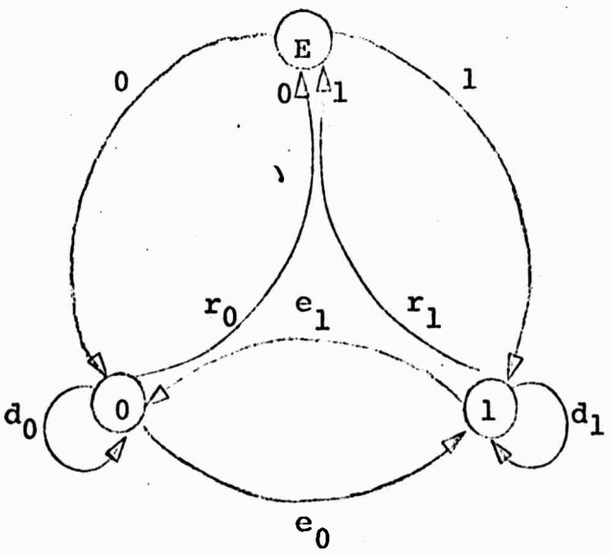
D1. We will construct and analyze a state machine which represents a binary channel. The channel may be in three states: empty, zero and one. From the empty state it may change to either the zero state or the one state.



At these states, one of three things may occur: return to the empty state: delay without error, and delay with error.



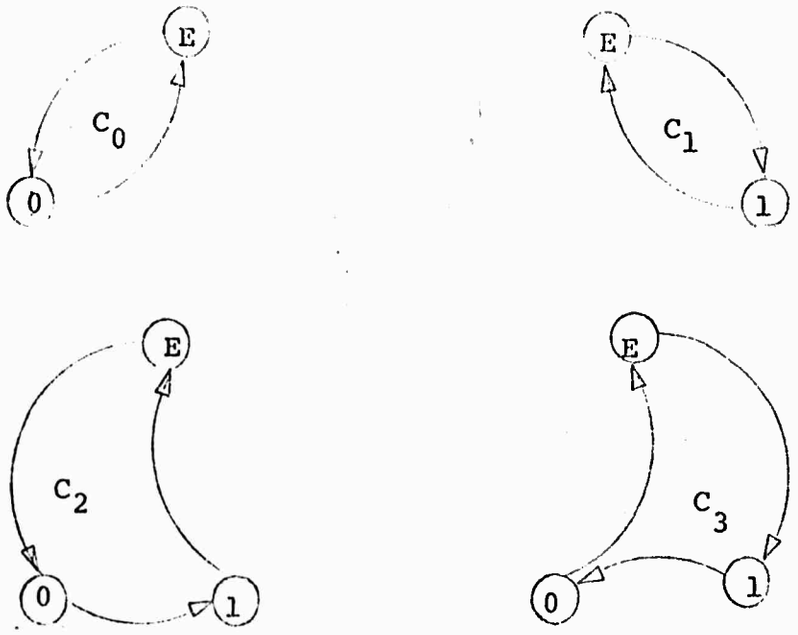
We will label each arc with an input character and label the arcs entering E with appropriate output characters.

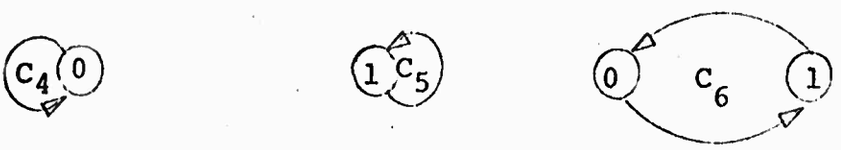


Using the methods described in the preceding sections we can determine where input information is output.

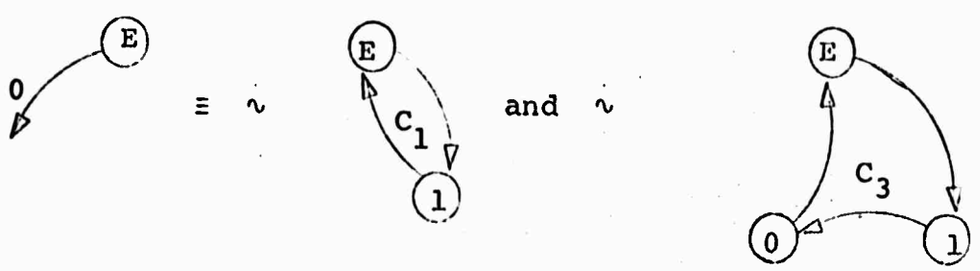
D2. To represent information quanta in this machine we will use circuits in the method described in section B.

The circuits of this state machine are:





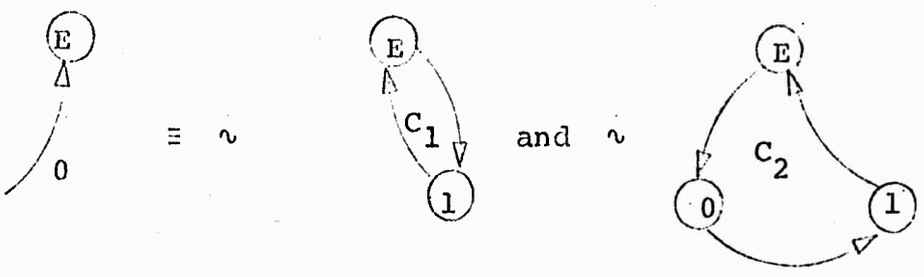
The input "0" at E is logically equivalent to the exclusion $\sim(C_1 \text{ or } C_3)$.



This exclusion means: "Neither C_1 nor C_3 is in the present D-graph."

The output "0" on arriving at E is logically

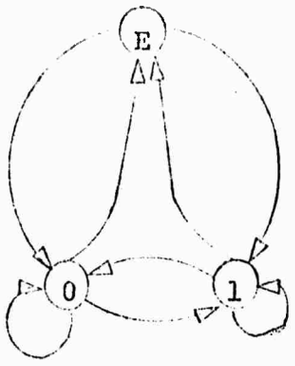
equivalent to the exclusion $\sim (C_1 \text{ or } C_2)$.



D3. We will show that, by our measure of information, the fraction of the information associated with the output "0" which is not part of the input "0" is equal to the probability that the 0 is an error.

The number of D-graphs rooted at E which contain circuits in $C_1 \cup C_2$ is 3 . One of these D-graphs contains C_2 . However, C_2 is not part of the input "0" . Thus the fraction of the output "0" which is not contained in the input "0" is $\frac{1}{3}$. Now assume E is entered by the 0 arc while the last exit from E is the 1 arc. Then the only possible D-graph is $\langle C_2 , E \rangle$. The probability of having left E by the 1 arc, given that we are at E and entered it by the 0 arc, is therefore $\frac{1}{3}$.

D4. We will now show how information quanta are input and output in a specific state sequence of the machine:

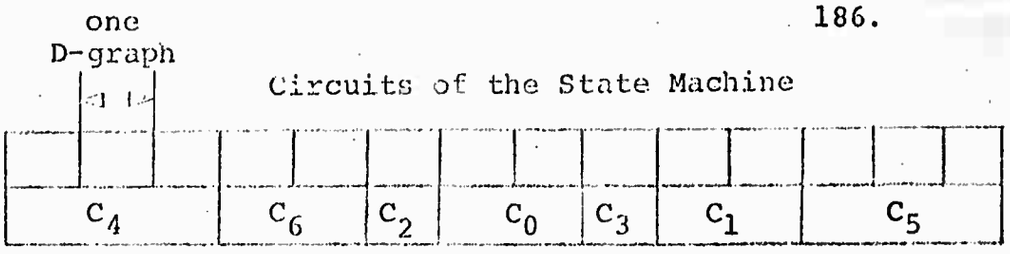


We will represent a state by a bar the length of which is proportional to the number of D-graphs rooted at that state. The bar is divided into sectors which represent circuits. The length of each sector is proportional to the number of D-graphs rooted at the state which contains that circuit. The circuit sets associated with the states are:

$$C(E) = \{C_0, C_1, C_2, C_3\}$$

$$C(0) = \{C_4, C_0, C_2, C_3, C_6\}$$

$$C(D) = \{C_5, C_1, C_2, C_3, C_6\}$$



An error free sequence

E C_2 C_0 C_3 C_1

I: ∿ (C_1, C_3) C_2 C_0

O: ∿ (C_3, C_4, C_6)

0 C_4 C_6 C_2 C_0 C_3

: ∿ (C_0, C_2, C_3, C_6) C_4

): ∿ $(C_0, C_2, C_3, C6)$

0 C_4 C_6 C_2 C_0 C_3

..: ∿ (C_4, C_6, C_2) C_0 C_3

: ∿ (C_2, C_1)

E C_2 C_0 C_3 C_1

A sequence with error

E C_2 C_0 C_3 C_1

: ∿ (C_1, C_3) C_2 C_0

(∿ (C_3, C_4, C_6)

0 C_4 C_6 C_2 C_0 C_3

1. ∿ (C_0, C_3, C_4) C_6 C_2

C. ∿ (C_3, C_1, C_5)

1 C_6 C_2 C_3 C_1 C_5

I: ∿ (C_3, C_5, C_6) C_2 C_1

O: ∿ (C_0, C_3)

E C_2 C_0 C_3 C_1

Note that in the error free sequence $\sim C_1$ is input at E and output at E while in the error sequence, $\sim C_1$ is input at E and output at 1 .

Furthermore, in the error free sequence $\sim C_0$ is never input, while in the error sequence $\sim C_0$ is input at 0 , (when the error occurs, in fact) and is output at E .

BIBLIOGRAPHY OF RELATED WORK

Holt, Anatol W. "Final Report of Information System Theory Project," Rome Air Development Center, Contract AF30(602)-4211. Applied Data Research, Inc., Princeton, N.J. February 1968. Available from the Department of Commerce Clearinghouse, Springfield, Va., as Report # AD 676 972.

Holt, Anatol W., S.O. Chagnon and R.M. Shapiro. "MEM-Theory (working papers) - A Mathematical Method for the Description and Analysis of Discrete, Finite Information Systems," Applied Data Research, Inc., Princeton, N.J. September 1965. Available from the Department of Commerce Clearinghouse, Springfield, Va., in two volumes:
V. I -- MEM-Theory, Technical Documentary Report #1 -- AD 626 819;
V. II -- Collected Research Papers -- AD 626 820.

Patil, Suhas S. "Macro-Modular Circuit Design" - Computation Structures Group Memo No. 40, Massachusetts Institute of Technology, Project MAC, Cambridge, Mass. May 1969.

Patil, Suhas S. "Macro-Modular Design of Asynchronous Circuits" - Computation Structures Group Memo No. 41, Massachusetts Institute of Technology, Project MAC, Cambridge, Mass. May 1969.

Patil, Suhas S. "n-server m-user Arbiter" - Computation Structures Group Memo No. 42, Massachusetts Institute of Technology, Project MAC, Cambridge, Mass. May 1969.

Patil, Suhas S. "A Micro-Modular Implementation of the Control Molues of Basic Macri-Modular Circuits" - Computation Structures Group Memo No. 43, Massachusetts Institute of Technology, Project MAC, Cambridge, Mass. October 1969.

Patil, Suhas S. "Communications Requirements for Determinacy of a System of Interconnected Determinate Systems" - Statement on Memo No. 44. Massachusetts Institute of Technology, Project MAC, Cambridge, Mass.

Petri, Carl Adam. "Communication with Automata." Translated by Clifford F. Greene, Jr.,

Shapiro, Robert M., Harry Saint et al. "A Handbook on File Structuring" - Vol. I. of the Final Technical Report, Rome Air Development Center, RADC-TR-69-313. Applied Data Research, Inc., Princeton, N.J. September 1969.

Shapiro, Robert M., and Harry Saint. " The Representation of Algorithms" - Vol. II. of the Final Technical Report, Rome Air Development Center, RADC-TR-69-313. Applied Data Reserach, Inc., Princeton, N.J. September 1969.

Also related, though less directly, is a large body of work to be found under such headings as Time Independence, Asynchronous Operation and Parallel Computation. Much work on such topics has recently been done at Massachusetts Institute of Technology, Washington University, U.C.L.A and Berkeley.

MASSACHUSETTS COMPUTER ASSOCIATES
A DIVISION OF APPLIED DATA RESEARCH, INC.
LAKESIDE OFFICE PARK WAKEFIELD, MASSACHUSETTS 01880 (617) 245-9540

AD-704 796

August 24, 1970

Gentlemen:

~~Please replace this copy of the Third Annual Technical Report covering Task Area II dated 22 June 1970 - 21 December 1970 with the copy you now have.~~ Pages 1, 5, and 6 have been changed and this will give you an updated report.

Thank you

Massachusetts Computer Associates, Inc.,
Division of Applied Data Research, Inc.

4

INTRODUCTION

This report covers work on Task Area II of the ARPA sponsored project "Research in Machine-Independent Software Programming" covering the six-month period ending on December 21, 1969. Although technically, this report is the sequel to "The Final Report for the Information System Theory Project"[1], it is self-contained.

This report includes the work of Dr. Anatol W. Holt and Mr. F. Commoner. During the period noted above, consultations were held with Mr. Robert M. Shapiro, Dr. Carl Adam Petri, Herr Hartmann Genrich and Dr. Shimon Even.

This introduction is divided into three parts. First, a summary of the objectives of the project as a whole; second a reported contract period; third, a view of things to come.

---

[1]See bibliography for this and other directly related work.

to the study of two simpler classes of structures -- marked graphs[1] and state transition diagrams. Both of these are special cases of occurrence systems. We have reason to hope that our developing ability to analyze these two classes will give us the tools with which to attack the analysis of systems which are Petri-net describable. Marked graphs and state transition diagrams isolate two aspects of system description from one another: the aspect which has to do with flow, and the aspect which has to do with function. The analysis of flow (marked graphs) shows where items flow and what other items they meet; the analysis of function shows the structure of the items and how they affect one another.

In the area of marked graphs effort was divided into two parts: semantics and mathematics. Here "Semantics" means developing techniques for expressing meanings about systems in marked graph form (Chapters II and III of this report). On the side of mathematics many theorems and algorithms were developed which have significant system interpretation (Chapters IV and V).

---

[1]The first serious study of marked graphs was undertaken by Hartmann Genrich, a colleague of Carl Adam Petri at GMD -- Gesellschaft für Mathematik und Datenverarbeitung, 5201 Birlinghoven, West Germany. He communicated with us on two occasions -- in June 1968 and June 1969. By June 1968 he had already proved a set of conditions to be necessary and sufficient for liveness and safety in marked graphs. (There exists a technical memorandum about this at the GMD -- Das Zollstationenproblem.) Various of our theorems, notably E4-E6 and F2-F6, generalize facts which we became aware of through Genrich -- the direction of generalization is usually the elimination of safety, or of liveness and safety, as hypotheses. Genrich's doctoral dissertation on marked graphs will become available later this year from the GMD, under the title Einfache Nicht-sequentielle Prozesse.

In the area of state transition analysis we developed a new technical concept of information which makes it possible to measure information quantities that flow in and out of a state machine, as well as identify the information content which flows in and out at different state transitions (Chapter VI). This work is, thus far, of theoretical rather than practical interest.

C. Things to Come

In regard to marked graphs, we are approaching the point where it will be useful and necessary to build a program package for the construction and analysis of such graphs. Without such a package we will not learn how to build marked graph representations of practical systems. An analogy can be made to computer programming. Marked graphs (and more generally, Petri nets) are to system description as computer code is to programming. In order to represent large problems, one first needs assembly techniques, and subsequently higher-level languages. Efforts to mechanize such assembly of big descriptions are worth making since there now exist analytic tools which will reveal interesting properties of the resulting descriptions.

Several next steps are indicated in moving toward the analysis of Petri nets. One direction is the direct extension of present marked graph results. Another direction is to form the appropriate connections between our existing work on state transition diagrams and marked graphs. Both of these directions will be pursued in the next period.

This report consists of an introduction to a series of technical documents describing specific research tasks performed during this reporting period.

These documents are currently in final stages of preparation and will be distributed as supplemental Special Technical Reports under the supporting contract.